```
UUU           UUU     AAAAAAAAA      FFFFFFFFFFFFFFF
UUU           UUU     AAAAAAAAA      FFFFFFFFFFFFFFF
UUU           UUU     AAAAAAAAA      FFFFFFFFFFFFFFF
UUU           UUU  AAA           AAA  FFF
UUU           UUU  AAA           AAA  FFF
UUU           UUU  AAA           AAA  FFF
UUU           UUU  AAA           AAA  FFF
UUU           UUU  AAA           AAA  FFF
UUU           UUU  AAA           AAA  FFF
UUU           UUU  AAA           AAA  FFFFFFFFFFFF
UUU           UUU  AAA           AAA  FFFFFFFFFFFF
UUU           UUU  AAA           AAA  FFFFFFFFFFFF
UUU           UUU  AAAAAAAAAAAAAAA  FFF
UUU           UUU  AAAAAAAAAAAAAAA  FFF
UUU           UUU  AAAAAAAAAAAAAAA  FFF
UUU           UUU  AAA           AAA  FFF
UUU           UUU  AAA           AAA  FFF
UUU           UUU  AAA           AAA  FFF
UUUUUUUUUUUUUUUU  AAA           AAA  FFF
UUUUUUUUUUUUUUUU  AAA           AAA  FFF
UUUUUUUUUUUUUUUU  AAA           AAA  FFF
```

```
_$2

Val
----
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0003
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
000
000
0000
000
000
000
000
```

FILEID**UAFMAIN

```
UU       UU    AAAAAA    FFFFFFFFFF  MM      MM    AAAAAA      IIIIII    NN       NN
UU       UU    AAAAAA    FFFFFFFFFF  MM      MM    AAAAAA      IIIIII    NN       NN
UU       UU  AA      AA  FF          MMMM  MMMM  AA      AA      II      NN       NN
UU       UU  AA      AA  FF          MMMM  MMMM  AA      AA      II      NN       NN
UU       UU  AA      AA  FF          MM  MM  MM  AA      AA      II      NNNN     NN
UU       UU  AA      AA  FF          MM  MM  MM  AA      AA      II      NNNN     NN
UU       UU  AA      AA  FFFFFFFF    MM      MM  AA      AA      II      NN  NN   NN
UU       UU  AA      AA  FFFFFFFF    MM      MM  AA      AA      II      NN  NN   NN
UU       UU  AAAAAAAAAA  FF          MM      MM  AAAAAAAAAA      II      NN    NNNN
UU       UU  AAAAAAAAAA  FF          MM      MM  AAAAAAAAAA      II      NN    NNNN
UU       UU  AA      AA  FF          MM      MM  AA      AA      II      NN      NN      ....
UU       UU  AA      AA  FF          MM      MM  AA      AA      II      NN      NN      ....
UUUUUUUUUU   AA      AA  FF          MM      MM  AA      AA    IIIIII    NN      NN      ....
UUUUUUUUUU   AA      AA  FF          MM      MM  AA      AA    IIIIII    NN      NN      ....

LL             IIIIII      SSSSSSSS
LL             IIIIII      SSSSSSSS
LL               II      SS
LL               II      SS
LL               II      SS
LL               II      SS
LL               II        SSSSSS
LL               II        SSSSSS
LL               II              SS
LL               II              SS
LL               II              SS
LL               II              SS
LLLLLLLLLL     IIIIII    SSSSSSSS
LLLLLLLLLL     IIIIII    SSSSSSSS
```

```
;     1          0001  0 module uafmain (main = start,
;     2          0002  0                  language (bliss32),
;     3          0003  0                  ident = 'V04-000',
;     4          0004  0                  addressing_mode (external=general, nonexternal=general)
;     5          0005  0                  ) =
;     6          0006  1 begin
;     7          0007  1
;     8          0008  1 !
;     9          0009  1 !**************************************************************************
;    10          0010  1 !*                                                                        *
;    11          0011  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                               *
;    12          0012  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
;    13          0013  1 !*  ALL RIGHTS RESERVED.                                                   *
;    14          0014  1 !*                                                                        *
;    15          0015  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    16          0016  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    17          0017  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    18          0018  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    19          0019  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    20          0020  1 !*  TRANSFERRED.                                                           *
;    21          0021  1 !*                                                                        *
;    22          0022  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    23          0023  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    24          0024  1 !*  CORPORATION.                                                           *
;    25          0025  1 !*                                                                        *
;    26          0026  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    27          0027  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
;    28          0028  1 !*                                                                        *
;    29          0029  1 !*                                                                        *
;    30          0030  1 !**************************************************************************
;    31          0031  1
;    32          0032  1 !++
;    33          0033  1 ! FACILITY:      System Management Utility Program
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT:
;    36          0036  1 !
;    37          0037  1 !       This program allows the system manager to maintain the user
;    38          0038  1 !       authorization file which contains usernames, passwords, quotas,
;    39          0039  1 !       and defaults.  The following functions are provided:
;    40          0040  1 !
;    41          0041  1 !               ADD - add a new user record to the authorization file (UAF)
;    42          0042  1 !               COPY - copy a user record, give copied record a new name
;    43          0043  1 !               DEFAULT - change a default value
;    44          0044  1 !               EXIT - exit program and update file
;    45          0045  1 !               HELP - ask for explanation
;    46          0046  1 !               LIST - complete list of records to file
;    47          0047  1 !               MODIFY - change one or more values for a user
;    48          0048  1 !               REMOVE - remove a user record
;    49          0049  1 !               RENAME - rename a user record (COPY; REMOVE)
;    50          0050  1 !               SHOW - display the values from a user record
;    51          0051  1 !
;    52          0052  1 ! ENVIRONMENT:
;    53          0053  1 !
;    54          0054  1 ! AUTHOR:        Henry M. Levy, CREATION DATE: 1-June-1977
;    55          0055  1 !
;    56          0056  1 ! MODIFIED BY:
;    57          0057  1 !
```

```
;    58          0058  1 ;        V03-024 JRL0036         John R. Lawson, Jr.     07-Aug-1984 17:08
;    59          0059  1 ;                Hide UPGRADE, DOWNGRADE, TMPJNL, PRMJNL privileges.  This
;    60          0060  1 ;                is a temporary work-around; it is marked in the right-hand
;    61          0061  1 ;                margin with !** in routine PRINT_PRIV.
;    62          0062  1 ;
;    63          0063  1 ;        V03-023 JRL0027         John R. Lawson, Jr.     25-Jul-1984 11:01
;    64          0064  1 ;                Add MODIFY/SYSTEM_PASSWORD=xxxxx to modify the system
;    65          0065  1 ;                password.
;    66          0066  1 ;
;    67          0067  1 ;        V03-022 JRL0029         John R. Lawson, Jr.     25-Jul-1984 10:47
;    68          0068  1 ;                Find better names for UAF$_NEWMSG10 and UAF$_NEWMSG15
;    69          0069  1 ;
;    70          0070  1 ;        V03-021 JRL0020         John R. Lawson, Jr.     09-Jul-1984 15:51
;    71          0071  1 ;                Bark when a proxy name is changed with the RENAME command.
;    72          0072  1 ;
;    73          0073  1 ;        V03-020 JRL0017         John R. Lawson, Jr.     02-Jul-1984 22:13
;    74          0074  1 ;                Modify GET_UAF_RECORD so that it does not get the system
;    75          0075  1 ;                password record.
;    76          0076  1 ;
;    77          0077  1 ;        V03-029 JRL0013         John R. Lawson, Jr.     02-Jul-1984 12:28
;    78          0078  1 ;                Display privileges of users in groups less than .EXE$GL_SYSUIC
;    79          0079  1 ;                as "ALL".
;    80          0080  1 ;
;    81          0081  1 ;        V03-028 JRL0010         John R. Lawson, Jr.     25-Jun-1984 15:56
;    82          0082  1 ;                Changed 'X'/'-' to '#'/'-' in primary/secondary access
;    83          0083  1 ;                display.
;    84          0084  1 ;
;    85          0085  1 ;        V03-027 JRL0008         John R. Lawson, Jr.     21-Jun-1984 14:00
;    86          0086  1 ;                Add support for the /PWDEXPIRED qualifier (pre-expired
;    87          0087  1 ;                password).
;    88          0088  1 ;
;    89          0089  1 ;        V03-026 JRL0006         John R. Lawson, Jr.     20-Jun-1984 12:28
;    90          0090  1 ;                Obliterate operator's console messages
;    91          0091  1 ;
;    92          0092  1 ;        V03-025 JRL0002         John R. Lawson, Jr.     15-Jun-1984 09:55
;    93          0093  1 ;                Change all internal messages to calls to LIB$SIGNAL and
;    94          0094  1 ;                the message u'ility -- place all messages in UAFMSG.MSG
;    95          0095  1 ;
;    96          0096  1 ;        V03-024 LY0494          Larry Yetto             11-JUN-1984 12:59
;    97          0097  1 ;                Fix noise error message comming from ADD/ID/USER=* caused
;    98          0098  1 ;                by a flag not properly being set
;    99          0099  1 ;
;   100          0100  1 ;        V03-023 MHB0150         Mark Bramhall           2-May-1984
;   101          0101  1 ;                Remove unused reference to TPARSE definitions.
;   102          0102  1 ;                Add DISPECONNECT flag.
;   103          0103  1 ;                Add security auditing for SYSUAF/NETUAF changes.
;   104          0104  1 ;
;   105          0105  1 ;        V03-022 LY0474          Larry Yetto             9-APR-1984 08:32
;   106          0106  1 ;                Zero the login failure and last login fields
;   107          0107  1 ;                on a copy operation.
;   108          0108  1 ;
;   109          0109  1 ;        V03-021 LY0466          Larry Yetto             22-MAR-1984 13:52
;   110          0110  1 ;                Add support for the rights data base functions
;   111          0111  1 ;
;   112          0112  1 ;        V03-020 ACG0397         Andrew C. Goldstein,    24-Feb-1984  23:21
;   113          0113  1 ;                Clean up display formatting
;   114          0114  1 ;
```

```
;  115     0115  1 ;        V03-019 ACG0397         Andrew C. Goldstein,    6-Feb-1984  16:27
;  116     0116  1 ;                Add DISREPORT to flags, clean up record locking
;  117     0117  1 ;
;  118     0118  1 ;        V03-018 ACG0388         Andrew C. Goldstein,    12-Jan-1984  19:21
;  119     0119  1 ;                Add command input to handle new UAF features;
;  120     0120  1 ;                general code cleanup
;  121     0121  1 ;
;  122     0122  1 ;        V03-017 ACG0385         Andrew C. Goldstein,    6-Jan-1984  18:28
;  123     0123  1 ;                V4 UAF format change; remove read-only under installed
;  124     0124  1 ;                SYSPRV feature; misc. code cleanups
;  125     0125  1 ;
;  126     0126  1 ;        V03-016 TMK0001         Todd M. Katz            10-Oct-1983
;  127     0127  1 ;                Add JTQUOTA (job-wide logical name table creation quota)
;  128     0128  1 ;                qualifier.
;  129     0129  1 ;
;  130     0130  1 ;        V03-015 LMP0153         L. Mark Pilant,         13-Sep-1983  11:57
;  131     0131  1 ;                Add minimal support for alphanumeric UICs.
;  132     0132  1 ;
;  133     0133  1 ;        014     JWT0105         Jim Teague      30-Mar-1983
;  134     0134  1 ;                Small changes to CLITABLES implementation.
;  135     0135  1 ;
;  136     0136  1 ;        013     JWT0104         Jim Teague      29-Mar-1983
;  137     0137  1 ;                Add CLITABLES qualifier.
;  138     0138  1 ;
;  139     0139  1 ;        012     WMC0001         Wayne Cardoza   15-Mar-1983
;  140     0140  1 ;                Add MAXDETACH qualifier.
;  141     0141  1 ;
;  142     0142  1 ;        011     JWT0097         Jim Teague      23-Feb-1983
;  143     0143  1 ;                Fix RENAME problem with proxy entries.
;  144     0144  1 ;
;  145     0145  1 ;        010     JWT0096         Jim Teague      08-Feb-1983
;  146     0146  1 ;                Log NETUAF changes to console, too.
;  147     0147  1 ;
;  148     0148  1 ;        009     JWT0087         Jim Teague      11-Jan-1983
;  149     0149  1 ;                Change SYSWSQUOTA for created UAFs to 350
;  150     0150  1 ;
;  151     0151  1 ;        008     JWT0082         Jim Teague      05-Jan-1983
;  152     0152  1 ;                Fix problem with LIST/PROXY.
;  153     0153  1 ;
;  154     0154  1 ;        007     JWT0079         Jim Teague      15-Dec-1982
;  155     0155  1 ;                Enlarge output field for BYTLM; reset pending
;  156     0156  1 ;                mail count for COPY operations.
;  157     0157  1 ;
;  158     0158  1 ;        006     JWT0072         Jim Teague      03-Dec-1982
;  159     0159  1 ;                Add global longword which can be patched to
;  160     0160  1 ;                enable/disable console logging of SYSUAF mods.
;  161     0161  1 ;
;  162     0162  1 ;        005     JWT0069         Jim Teague      24-Nov-1982
;  163     0163  1 ;                Allow redefinition of sys$output.
;  164     0164  1 ;
;  165     0165  1 ;        004     JWT0057         Jim Teague      21-Sep-1982
;  166     0166  1 ;                Add a message to tell whether or not NETUAF was
;  167     0167  1 ;                modified.
;  168     0168  1 ;
;  169     0169  1 ;        003     JWT0042         Jim Teague      15-Jul-1982
;  170     0170  1 ;                Make SYSUAF.LIS and NETUAF.LIS world noread.
;  171     0171  1 ;
```

```
;  172          0172  1 !          002     JWT0036         Jim Teague      08-Jun-1982
;  173          0173  1 !                  Add full wildcarding to show/proxy
;  174          0174  1 !
;  175          0175  1 !          001     JWT0022         Jim Teague      17-Mar-1982
;  176          0176  1 !                  Fix bug that caused failure to reparse command line for
;  177          0177  1 !                  wildcard modifications.  List default device on its own
;  178          0178  1 !                  line for show/full.
;  179          0179  1 !
;  180          0180  1 !--
```

```
;  182     0181  1 !
;  183     0182  1 ! Require files
;  184     0183  1 !
;  185     0184  1 require
;  186     0185  1         'lib$:uafreq';
;  187     0281  1
;  188     0282  1 !
;  189     0283  1 ! INCLUDE FILES:
;  190     0284  1 !
;  191     0285  1 library 'SYS$LIBRARY:LIB.L32';
;  192     0286  1
;  193     0287  1 !
;  194     0288  1 ! TABLE OF CONTENTS:
;  195     0289  1 !
;  196     0290  1
;  197     0291  1 forward routine
;  198     0292  1         start,                          ! controlling code
;  199     0293  1         setup           : novalue,      ! open initial files
;  200     0294  1         add_uaf         : novalue,      ! insert new user record
;  201     0295  1         add_proxy       : novalue,      ! insert new proxy record
;  202     0296  1         remote_parse,                   ! parses "node: : remoteuser"
;  203     0297  1         copy_uaf,                       ! copy user record
;  204     0298  1         create_proxy    : novalue,      ! create NETUAF.DAT proxy file
;  205     0299  1         modify_uaf      : novalue,      ! update user record(s)
;  206     0300  1         modify_rec,                     ! update a user record action routine
;  207     0301  1         remove_uaf      : novalue,      ! remove username from file
;  208     0302  1         remove_proxy    : novalue,      ! remove a proxy record
;  209     0303  1         rename_uaf      : novalue,      ! rename user record
;  210     0304  1         adjust_proxy    : novalue,      ! implicitly remove/update proxy record
;  211     0305  1         default_uaf     : novalue,      ! change default record
;  212     0306  1         list_proxy      : novalue,      ! list proxy entries in NETUAF.LIS
;  213     0307  1         list_uaf        : novalue,      ! list file routine
;  214     0308  1         show_user_uaf   : novalue,      ! display user record
;  215     0309  1         show_proxy      : novalue,      ! display proxy record at terminal
;  216     0310  1         locate_proxy,                   ! access given proxy record(s)
;  217     0311  1         get_proxy_record,               ! read single proxy record
;  218     0312  1         display_proxy   : novalue,      ! format and output a proxy entry
;  219     0313  1         wild_user,                      ! user wild card routine
;  220     0314  1         display_brief,                  ! writes a brief user display
;  221     0315  1         classify_priv,                  ! classifies contents of priv vector
;  222     0316  1         display_full,                   ! writes the full user display
;  223     0317  1         display_hours   : novalue,      ! display hourly restrictions
;  224     0318  1         convert_time    : novalue,      ! convert time value to string
;  225     0319  1         print_priv      : novalue,      ! print privilege bits
;  226     0320  1         build_ini_recs  : novalue,      ! build initial file records for default
;  227     0321  1                                         ! and system manager
;  228     0322  1         get_user_record,                ! get username and lookup record
;  229     0323  1         locate_user,                    ! lookup user record in UAF
;  230     0324  1         get_uaf_record,                 ! routine to deal with record locking
;  231     0325  1         get_cmd_line,                   ! input user command line
;  232     0326  1         ask             : novalue,      ! prompt terminal for input
;  233     0327  1         fmt_sys_msg     : novalue,      ! output system message file message
;  234     0328  1         faoout,                         ! output formatted message
;  235     0329  1         help_uaf        : novalue,      ! help routine
;  236     0330  1         exit_uaf        : novalue,      ! normal exit routine
;  237     0331  1         SIGNAL_SYNTAX   : novalue,      ! missing qualifier
;  238     0332  1         acc$exit        : novalue,      ! exit and cleanup routine
```

```
;  239     0333  1         uaf$mod_sys_pwd : novalue,      . modify the system password
;  240     0334  1         security_audit  : novalue;      . Perform a security audit
;  241     0335  1
;  242     0336  1 linkage
;  243     0337  1         fmg_match = jsb (register = 2, register = 3, register = 4,
;  244     0338  1                          register = 5) : notused (10, 11);
;  245     0339  1
;  246     0340  1 !
;  247     0341  1 ! EXTERNAL REFERENCES:
;  248     0342  1 !
;  249     0343  1
;  250     0344  1 external literal
;  251     0345  1         cli$_bufovf,
;  252     0346  1         cli$_noclint;
;  253     0347  1
;  254     0348  1
;  255     0349  1 external routine
;  256     0350  1         lbr$output_help,
;  257     0351  1         lib$get_foreign,
;  258     0352  1         lib$get_input,
;  259     0353  1         lib$put_output,
;  260     0354  1         fmg$match_name  : fmg_match,
;  261     0355  1         cli$dcl_parse,
;  262     0356  1         cli$dispatch,
;  263     0357  1         cli$present,
;  264     0358  1         cli$get_value,
;  265     0359  1         update_record,                  ! modify all specified fields
;  266     0360  1         parse_wild,                     ! parses a wildcarded user specification
;  267     0361  1         lgi$hpwd,                       ! hash password routine
;  268     0362  1         uaf$add_ident_recbuf,
;  269     0363  1         uaf$build_holder,
;  270     0364  1         uaf$find_uic ,
;  271     0365  1         uaf$remove_ident_recbuf,
;  272     0366  1         uaf$write_rights;
;  273     0367  1
;  274     0368  1 external
;  275     0369  1         EXE$GL_SYSUIC   : long ,
;  276     0370  1         rdb_header_flag : byte ,
;  277     0371  1         rdb_list_flag   : byte ,
;  278     0372  1         attributes      : long ,
;  279     0373  1         holder          : $bblock[8] ,
;  280     0374  1         ident           : $bblock [4] ,
;  281     0375  1         authorize_commands,             ! AUTHORIZE command parse tables
;  282     0376  1         prv$ab_names;                   ! address of privilege name table
;  283     0377  1
;  284     0378  1
;  285     0379  1 !
;  286     0380  1 ! MACROS:
;  287     0381  1 !
;  288     0382  1 macro
;  289     0383  1
;  290   M 0384  1     namelen (x, y) =
;  291   M 0385  1         begin
;  292   M 0386  1         builtin
;  293   M 0387  1             locc;
;  294   M 0388  1         register
;  295   M 0389  1             r0 = 0;
```

```
;  296        M 0390  1          locc (%ref(' '), %ref(x), y; r0);
;  297        M 0391  1          x = .r0
;  298          0392  1          end%,
;  299          0393  1
;  300        M 0394  1      cstring[] = (uplit byte (%charcount (%string (%remaining)),
;  301          0395  1                   %string (%remaining)))%,
;  302          0396  1
;  303          0397  1      fatal[] = (fmt_sys_msg (%remaining); acc$exit ())%,
;  304          0398  1  !
;  305          0399  1  ! Macros to check for success or failure from RMS
;  306          0400  1  !
;  307          0401  1      rmsbad (string) = (not (rmserr = string)) %,
;  308          0402  1      rmsok (string) = (rmserr = string) %,
;  309          0403  1  !
;  310          0404  1  ! Macros to set up and write an FAO string.
;  311          0405  1  !
;  312        M 0406  1      faomac (faomsg)[] =
;  313        M 0407  1          begin
;  314        M 0408  1          faodsc[dsc$w_length] = . (faomsg)<0,8>;
;  315        M 0409  1          faodsc[dsc$a_pointer] = (faomsg) + 1;
;  316        M 0410  1          $fao (faodsc, rabptr[rab$w_rsz], disdsc $comma (%remaining)
;  317        M 0411  1                           %remaining);
;  318        M 0412  1          $put (rab = .rabptr);
;  319          0413  1          end %,
;  320          0414  1
;  321          0415  1      $comma[] =, %,
;  322          0416  1
;  323        M 0417  1      output_null =
;  324        M 0418  1          begin
;  325        M 0419  1          rabptr[rab$w_rsz] = 0;
;  326        M 0420  1          $put (rab = .rabptr);
;  327          0421  1          end %,
;  328          0422  1
;  329          0423  1  !
;  330          0424  1  ! Macro to create string descriptor for command parameters and
;  331          0425  1  !       qualifiers
;  332          0426  1  !
;  333        M 0427  1      sd[a] =
;  334          0428  1          bind %name ('SD_',a) = $descriptor (a)%;
;  335          0429  1
;  336        P 0430  1       sd (
;  337        P 0431  1          'token1',       'token2',       'brief',        'full',
;  338        P 0432  1          'add_identifier',               'remove_identifier',
;  339        P 0433  1          'modify_identifier'
;  340          0434  1  );
;  341          0435  1
;  342          0436  1  field
;  343          0437  1      descr_fields =                      ! Define the fields for a DESCRIPTOR
;  344          0438  1          set
;  345          0439  1          length  = [dsc$w_length],
;  346          0440  1          dtype   = [dsc$b_dtype],
;  347          0441  1          class   = [dsc$b_class],
;  348          0442  1          pointer = [dsc$a_pointer]
;  349          0443  1          tes;
;  350          0444  1
;  351          0445  1  macro
;  352        M 0446  1      statdesc =
```

```
;  353        M 0447  1          $bbblock [dsc$k_s_bln] field (descr_fields)
;  354        M 0448  1                    preset ( [length]  = 0,
;  355        M 0449  1                             [dtype]   = dsc$k_dtype_t,
;  356        M 0450  1                             [class]   = dsc$k_class_s,
;  357          0451  1                             [pointer] = 0)%;
;  358          0452  1
;  359          0453  1 macro
;  360        M 0454  1     qualstr_desc (string) =
;  361        M 0455  1          $bbblock [dsc$k_s_bln] field (descr_fields)
;  362        M 0456  1                    preset ( [length]  = (%charcount(string)),
;  363        M 0457  1                             [dtype]   = dsc$k_dtype_t,
;  364        M 0458  1                             [class]   = dsc$k_class_s,
;  365          0459  1                             [pointer] = (uplit byte (%string(string))))%;
;  366          0460  1 own
;  367          0461  1     sd_attribresource   : qualstr_desc ('attributes.resource') ;
;  368          0462  1
;  369          0463  1 !
;  370          0464  1 ! EQUATED SYMBOLS:
;  371          0465  1 !
;  372          0466  1 literal
;  373          0467  1          cmdbufmax       = 508,          ! maximum command length
;  374          0468  1          false           = 0,            ! logical false
;  375          0469  1          true            = 1,            ! logical true
;  376          0470  1          update_records  = 0,            ! flag for proxy file adjustment
;  377          0471  1          remove_records  = 1,            !      "     "     "      "
;  378          0472  1          copy_flag       = 1,            ! used in routine copy_uaf
;  379          0473  1          rename_flag     = 1,            ! used in routine rename_uaf
;  380          0474  1          byte_length     = 8,            ! bits per byte
;  381          0475  1          word_length     = 16,           ! bits per word
;  382          0476  1          long_length     = 32,           ! bits per longword
;  383          0477  1          retry_rlk       = 8,            ! number of retries for a locked record
;  384          0478  1          sleep_rlk       = 500,          ! ms to sleep before retrying
;  385          0479  1          cr              = 13,
;  386          0480  1          lf              = 10,
;  387          0481  1          zero            = 0,
;  388          0482  1          cmdbuflen       = 1024;         ! size of user command buffer
;  389          0483  1
;  390          0484  1 global literal
;  391          0485  1          encrypt         = uaf$c_purdy_v, ! encryption algorithm to use
;  392          0486  1          disbuflen       = 132;          ! size of display file output buffer
;  393          0487  1
;  394          0488  1 bind
;  395          0489  1          sysuaf_string   = uplit byte ('SYSUAF'),
;  396          0490  1          netuaf_string   = uplit byte ('NETUAF'),
;  397          0491  1          mod_act_dsc     = $descriptor ('modified'),
;  398          0492  1          add_act_dsc     = $descriptor ('added'),
;  399          0493  1          rem_act_dsc     = $descriptor ('removed'),
;  400          0494  1          fao_lin_dsc     = $descriptor ('PID=!XL !AS !AS !AS record !AS on !%D'),
;  401          0495  1          dbl_colon       = $descriptor ('::'),
;  402          0496  1 !
;  403          0497  1 ! Define the system delta time to sleep before retrying to GET a locked record.
;  404          0498  1 !
;  405          0499  1          wakedelta       = uplit long (-10*1000*sleep_rlk,-1),
;  406          0500  1 !
;  407          0501  1 ! Default values for authorization file record.  These values are
;  408          0502  1 ! only used when a new authorizatior file is created.  If the file
;  409          0503  1 ! already exists, the default values are read from the first file
```

```
;  410          0504  1 ! record.
;  411          0505  1 !
;  412          0506  1         defuser         = cstring ('DEFAULT'),  ! default username
;  413          0507  1         defpass         = cstring ('USER'),     ! default password
;  414          0508  1         defclitabl      = cstring ('DCLTABLES'), ! default CLI tables
;  415          0509  1         defact          = cstring (''),         ! default account name
;  416          0510  1         defcli          = cstring ('DCL'),      ! default command interpreter
;  417          0511  1         defowner        = cstring (''),         ! owner's name
;  418          0512  1         deflgicmd       = cstring (''),         ! default login command file
;  419          0513  1         defgrp          = %o'200',              ! default group
;  420          0514  1         defmem          = %o'200',              ! default member
;  421          0515  1         defdir          = cstring ('[USER]'),   ! default directory name
;  422          0516  1         defdev          = cstring (''),         ! default device name
;  423          0517  1         defbiolm        = 6,                    ! default buffered I/O limit
;  424          0518  1         defbytlm        = 4096,                 ! default buffered I/O buffer space
;  425          0519  1         defdiolm        = 6,                    ! default direct I/O limit
;  426          0520  1         deffillm        = 20,                   ! default open file limit
;  427          0521  1         defflags        = 0,                    ! default flag bits
;  428          0522  1         deftqcnt        = 10,                   ! default time queue entries
;  429          0523  1         defprccnt       = 2,                    ! default subprocess count
;  430          0524  1         defpri          = 4,                    ! default process priority
;  431          0525  1         defquepri       = 4,                    ! default queue priority
;  432          0526  1         defwsquota      = 200,                  ! default working set limit
;  433          0527  1         defdfwscnt      = 150,                  "default" working set default size
;  434          0528  1         defwsextent     = 500,                  ! default working set extent
;  435          0529  1         defcputim       = 0,                    ! default CPU time quota
;  436          0530  1         defastlm        = 10,                   ! default AST queue limit
;  437          0531  1         defpgflquota    = 10000,                ! default paging file limit in pages
;  438          0532  1         defenqlm        = 10,                   ! default enqueue limit
;  439          0533  1         defpbytlm       = 0,                    ! default paged buffer I/O limit
;  440          0534  1         defshrfillm     = 0,                    ! default shared file limit
;  441          0535  1         defmaxjobs      = 0,                    ! default maximum concurrent jobs
;  442          0536  1         defmaxacctjobs  = 0,                    ! default maximum concurrent group jobs
;  443          0537  1         defmaxdetach    = 0,                    ! default maximum detached processes
;  444          0538  1         defjtquota      = 1024,                 ! default job-wide logical table quota
;  445          0539  2         defprimedays    = (                     ! Sat, Sun are default non-prime days
;  446          0540  2                           (1 ^ $bitposition (uaf$v_saturday)) or
;  447          0541  3                           (1 ^ $bitposition (uaf$v_sunday))
;  448          0542  1                           ),
;  449          0543  1         defhours        = 0,                    ! default all hours to allow access
;  450          0544  1                 defpriv = uplit (               ! default privilege vector
;  451          0545  2                           (
;  452          0546  2                           (1 ^ $bitposition (prv$v_netmbx)) or
;  453          0547  3                           (1 ^ $bitposition (prv$v_tmpmbx))
;  454          0548  1                           ), 0),
;  455          0549  1         defpwdlength    = 6,                    ! default min password length
;  456          0550  1         defpwdlife      = uplit (0,0),          ! default password lifetime
;  457          0551  1
;  458          0552  1 !
;  459          0553  1 ! The following are the default values for the SYSTEM user.  When
;  460          0554  1 ! a new file is created, a system manager record is created.
;  461          0555  1 !
;  462          0556  1         sysuser         = cstring ('SYSTEM'),
;  463          0557  1         syspass         = cstring ('MANAGER'),
;  464          0558  1         sysclitabl      = cstring ('DCLTABLES'),
;  465          0559  1         sysact          = cstring ('SYSTEM'),
;  466          0560  1         syscli          = cstring ('DCL'),
```

```
;  467     0561  1          sysowner        = cstring ('SYSTEM MANAGER'),
;  468     0562  1          syslgicmd       = cstring (''),
;  469     0563  1          sysgrp          = %o'1',
;  470     0564  1          sysmem          = %o'4',
;  471     0565  1          sysdir          = cstring ('[SYSMGR]'),
;  472     0566  1          sysdev          = cstring (''),
;  473     0567  1          sysbiolm        = 12,
;  474     0568  1          sysbytlm        = 20480,
;  475     0569  1          sysdiolm        = 12,
;  476     0570  1          sysfillm        = 20,
;  477     0571  1          sysflags        = 0,
;  478     0572  1          systqcnt        = 20,
;  479     0573  1          sysprccnt       = 10,
;  480     0574  1          syspri          = 4,
;  481     0575  1          sysquepri       = 4,
;  482     0576  1          syswsquota      = 350,
;  483     0577  1          syswsextent     = 1024,
;  484     0578  1          sysdfwscnt      = 150,
;  485     0579  1          syscputim       = 0,
;  486     0580  1          sysastlm        = 20,
;  487     0581  1          syspgflquota    = 10000,
;  488     0582  1          sysenqlm        = 20,
;  489     0583  1          syspbytlm       = 0,
;  490     0584  1          sysshrfillm     = 0,
;  491     0585  1          sysmaxjobs      = 0,
;  492     0586  1          sysmaxdetach    = 0,
;  493     0587  1          sysjtquota      = 1024,
;  494     0588  1          sysmaxacctjobs  = 0,
;  495     0589  2          sysprimedays    = (                     ! Sat, Sun are default non-prime days
;  496     0590  2                            (1 ^ $bitposition (uaf$v_saturday)) or
;  497     0591  3                            (1 ^ $bitposition (uaf$v_sunday))
;  498     0592  1                            ),
;  499     0593  1          syshours        = 0,
;  500     0594  1          syspriv         = uplit (rep 2 of (%x'FFFFFFFF')),
;  501     0595  1          syspwdlength    = 8,                    ! default min password length
;  502     0596  1          syspwdlife      = uplit (0,0);          ! default password lifetime
;  503     0597  1
;  504     0598  1 !
;  505     0599  1 ! GLOBAL STORAGE - must be before OWN for initialization purposes
;  506     0600  1 !
;  507     0601  1 global
;  508     0602  1          disbuf          : vector [disbuflen, byte], ! display buffer
;  509     0603  1          disdsc          : block [8, byte] initial (disbuflen, disbuf) ;
;  510     0604  1
;  511     0605  1 !
;  512     0606  1 ! OWN STORAGE:
;  513     0607  1 !
;  514     0608  1 own
;  515     0609  1          username_buf    : block [uaf$s_username, byte],
;  516     0610  1          pcb_sts         : bitvector [32],
;  517     0611  1          pid,
;  518     0612  1          username_dsc    : vector [2] initial (0, username_buf),
;  519     0613  1          recname_dsc     : vector [2],
;  520     0614  1          file_dsc        : vector [2] initial (6),
;  521     0615  1          mod_default,                    ! DEFAULT record being modified by MODIFY command
;  522     0616  1          modify_flag     : long,         ! SYSUAF modified
;  523     0617  1          netuaf_modified,                ! NETUAF modified
```

```
;  54          0618  1          rename_ph2      : byte initial (false),
;  55          0619  1          oldUserLen      : long,
;  56          0620  1          oldUserName     : vector  [uaf$s_username,byte],
;  57          0621  1          newUserLen      : long,
;  58          0622  1          newUserName     : vector  [uaf$s_username,byte],
;  59          0623  1          cmdbuf          : vector [cmdbuflen, byte], ! command buffer
;  60          0624  1          default_size    : word,
;  61          0625  1          default_record  : block [uaf$c_length, byte], ! default record held here
;  62          0626  1          pwddsc          : block [8, byte],          ! password descriptor
;  63          0627  1          insize          : long,                     ! number of input chars
;  64          0628  1          brief_flag      : long,                     ! display option
;  65          0629  1          full_flag       : long,                     ! display option
;  66          0630  1          header_flag     : long,                     ! output header or not?
;  67          0631  1
;  68          0632  1
;  69        P 0633  1          infab   : $fab (                ! FAB for terminal I/O
;  70        P 0634  1                           fac = (get, put),
;  71        P 0635  1                           rat = cr,
;  72        P 0636  1                           fnm = 'SYS$INPUT'
;  73          0637  1                           ),
;  74          0638  1
;  75        P 0639  1          inrab   : $rab (                ! RAB for terminal I/O
;  76        P 0640  1                           rac = seq,
;  77        P 0641  1                           rop = pmt,
;  78        P 0642  1                           fab = infab
;  79          0643  1                           ),
;  80          0644  1
;  81        P 0645  1          outfab  : $fab (
;  82        P 0646  1                           fac = (put),
;  83        P 0647  1                           rat = cr,
;  84        P 0648  1                           fnm = 'SYS$OUTPUT'
;  85          0649  1                           ),
;  86          0650  1
;  87          0651  1          lstnam  : $nam (),              ! needed for the DLT option
;  88          0652  1
;  89        P 0653  1          lstpro  : $xabpro (
;  90        P 0654  1                           pro = (rwd,rwd,rw)
;  91          0655  1                           ),
;  92          0656  1
;  93        P 0657  1          lstfab  : $fab (                ! FAB for UAF listing file
;  94        P 0658  1                           fac = put,
;  95        P 0659  1                           rat = cr,
;  96        P 0660  1                           fnm = 'SYSUAF.LIS',
;  97        P 0661  1                           shr = nil,
;  98        P 0662  1                           org = seq,
;  99        P 0663  1                           rfm = var,
; 100        P 0664  1                           mrs = disbuflen,
; 101        P 0665  1                           nam = lstnam,
; 102        P 0666  1                           xab = lstpro
; 103          0667  1                           ),
; 104          0668  1
; 105        P 0669  1          lstrab  : $rab (                ! RAB for UAF listing file
; 106        P 0670  1                           rac = seq,
; 107        P 0671  1                           rbf = disbuf,
; 108        P 0672  1                           fab = lstfab
; 109          0673  1                           ),
; 110          0674  1
```

```
;  581          0675  1          nlstnam : $nam (),
;  582          0676  1
;  583        P 0677  1          nlstpro : $xabpro (
;  584        P 0678  1                            pro = (rwd,rwd,rw)
;  585          0679  1                            ),
;  586          0680  1
;  587        P 0681  1          nlstfab : $fab (                    ! FAB for NETUAF listing file
;  588        P 0682  1                            fac = put,
;  589        P 0683  1                            rat = cr,
;  590        P 0684  1                            fnm = 'NETUAF.LIS',
;  591        P 0685  1                            shr = nil,
;  592        P 0686  1                            org = seq,
;  593        P 0687  1                            rfm = var,
;  594        P 0688  1                            mrs = disbuflen,
;  595        P 0689  1                            nam = nlstnam,
;  596        P 0690  1                            xab = nlstpro
;  597          0691  1                            ),
;  598          0692  1
;  599        P 0693  1          nlstrab : $rab (                    ! RAB for NETUAF listing file
;  600        P 0694  1                            rac = seq,
;  601        P 0695  1                            rbf = disbuf,
;  602        P 0696  1                            fab = nlstfab
;  603          0697  1                            ),
;  604          0698  1
;  605        P 0699  1          uafkey2 : $xabkey (                 ! XAB for User number key
;  606        P 0700  1                            kref = 2,         ! alternate key
;  607        P 0701  1                            pos0 = $byteoffset (uaf$w_mem),
;  608        P 0702  1                            siz0 = 2,
;  609        P 0703  1                            dtp = bn2,
;  610        P 0704  1                            flg = (chg,dup)
;  611          0705  1                            ),
;  612          0706  1
;  613        P 0707  1          uafkey1 : $xabkey (                 ! XAB for Group number key
;  614        P 0708  1                            kref = 1,         ! alternate key
;  615        P 0709  1                            pos0 = $byteoffset (uaf$l_uic),
;  616        P 0710  1                            siz0 = 4,
;  617        P 0711  1                            dtp = bn4,
;  618        P 0712  1                            flg = (chg,dup),
;  619        P 0713  1                            nxt = uafkey2
;  620          0714  1                            ),
;  621          0715  1
;  622        P 0716  1          uafkey0 : $xabkey (                 ! XAB for USERNAME key
;  623        P 0717  1                            kref = 0,         ! primary key
;  624        P 0718  1                            pos0 = $byteoffset (uaf$t_username),
;  625        P 0719  1                            siz0 = uaf$s_username,
;  626        P 0720  1                            nxt = uafkey1
;  627          0721  1                            ),
;  628          0722  1
;  629        P 0723  1          uafpro  : $xabpro (                 ! XAB for file protection
;  630        P 0724  1                            pro = (rwed, rwed), ! deny world access
;  631        P 0725  1                            nxt = uafkey0
;  632          0726  1                            ),
;  633          0727  1
;  634        P 0728  1          uaffab  : $fab (                    ! FAB for work file
;  635        P 0729  1                            fop = cif,
;  636        P 0730  1                            fac = (get, put, del, upd),
;  637        P 0731  1                            fnm = 'SYSUAF',
```

```
;  638        P 0732  1                        dnm = '.DAT',
;  639        P 0733  1                        shr = (get, put, del, upd),
;  640        P 0734  1                        org = idx,
;  641        P 0735  1                        rfm = var,
;  642        P 0736  1                        mrs = uaf$c_length,
;  643        P 0737  1                        alq = 10,
;  644        P 0738  1                        deq = 10,
;  645        P 0739  1                        xab = uafpro
;  646           0740  1                        ),
;  647           0741  1 !
;  648           0742  1 ! FAB for NETUAF Proxy Login File
;  649           0743  1 !
;  650           0744  1 !
;  651        P 0745  1        nafkey1 : $xabkey (
;  652        P 0746  1                        kref = 1,
;  653        P 0747  1                        pos0 = $byteoffset (naf$t_localuser),
;  654        P 0748  1                        siz0 = naf$s_localuser,
;  655        P 0749  1                        flg = (chg,dup)
;  656           0750  1                        ),
;  657           0751  1
;  658        P 0752  1        nafkey0 : $xabkey (
;  659        P 0753  1                        kref = 0,
;  660        P 0754  1                        pos0 = $byteoffset (naf$t_remname),
;  661        P 0755  1                        siz0 = naf$s_remname,
;  662        P 0756  1                        nxt = nafkey1
;  663           0757  1                        ),
;  664           0758  1
;  665        P 0759  1        nafpro  : $xabpro (
;  666        P 0760  1                        pro = (rwed, rwed),
;  667        P 0761  1                        nxt = nafkey0
;  668           0762  1                        ),
;  669           0763  1
;  670        P 0764  1        naffab  : $fab (                ! FAB for NETUAF
;  671        P 0765  1                        fop = cif,
;  672        P 0766  1                        fac = (get, put, del, upd),
;  673        P 0767  1                        fnm = 'NETUAF',
;  674        P 0768  1                        dnm = '.DAT',
;  675        P 0769  1                        shr = (get, put, del, upd),
;  676        P 0770  1                        org = idx,
;  677        P 0771  1                        rfm = fix,
;  678        P 0772  1                        mrs = naf$c_length,
;  679        P 0773  1                        alq = 10,
;  680        P 0774  1                        deq = 10,
;  681        P 0775  1                        xab = nafpro
;  682           0776  1                        );
;  683           0777  1
;  684           0778  1 !
;  685           0779  1 ! GLOBAL STORAGE:
;  686           0780  1 !
;  687           0781  1 global
;  688           0782  1          faodsc          : block [8, byte],      ! gen'l purpose fao string desc
;  689           0783  1          rabptr          : ref block [rab$c_bln, byte], ! RAB for output
;  690           0784  1          uaf$gq_sysuaff  : block [nsa_s_sysuaff, byte], ! SYSUAF auditing flags
;  691           0785  1          uaf$gl_ctlmsk   : block [2],            ! Control mask for AUTHORIZE
;  692           0786  1          by_account      : long initial (false), ! processing by account
;  693           0787  1          match_token     : vector [naf$s_remname+2, byte], ! Saved match token
;  694           0788  1          match_tokenlen  : long,
```

```
;  695          0789  1          wild_netuser,                          ! wildcard access to proxy entries
;  696          0790  1          call_count      : long initial (0),    ! counter for reprocessing cmd line
;  697          0791  1          tokendsc        : block  [8, byte] preset ( [dsc$b_class]=dsc$k_class_d),
;  698          0792  1          cmdlindsc       : block  [8, byte],
;  699          0793  1          netuaf_exists,                         ! A self-explanatory flag...
;  700          0794  1          rdb_exists      : long,
;  701          0795  1          rmserr          : long,                ! save rms error codes here
;  702          0796  1          rightslist_modified : byte,            ! RIGHTSLIST modified flag
;  703          0797  1  !
;  704          0798  1  ! Record buffer for file I/O.  Records are generally read into RECBUF,
;  705          0799  1  ! modified, and output.
;  706          0800  1  !
;  707          0801  1          recbuf          : block [uaf$c_length, byte],
;  708          0802  1          netbuf          : block [naf$c_length, byte],
;  709          0803  1  !
;  710          0804  1  ! UAFRAB is global to allow UPDATE_RECORD to modify RAB$W_RSZ.
;  711          0805  1  !
;  712        P 0806  1          uafrab  : $rab (! RAB for work file
;  713        P 0807  1                           krf = 0,
;  714        P 0808  1                           kbf = recbuf [uaf$t_username],
;  715        P 0809  1                           ksz = uaf$s_username,
;  716        P 0810  1                           usz = uaf$c_length,
;  717        P 0811  1                           fab = uaffab
;  718          0812  1                           ),
;  719          0813  1  !
;  720          0814  1  ! RAB for NETUAF Proxy Login File
;  721          0815  1  !
;  722        P 0816  1          nafrab  : $rab (
;  723        P 0817  1                           krf = 0,
;  724        P 0818  1                           kbf = netbuf [naf$t_remname],
;  725        P 0819  1                           ksz = naf$s_remname,
;  726        P 0820  1                           usz = naf$c_length,
;  727        P 0821  1                           rsz = naf$c_length,
;  728        P 0822  1                           rac = key,
;  729        P 0823  1                           fab = naffab
;  730          0824  1                           ),
;  731          0825  1
;  732          0826  1
;  733          0827  1          time_buf        : block [8,byte],              !  current system time
;  734          0828  1          pwd_flag        : long,                ! Password default flag
;  735          0829  1
;  736        P 0830  1          outrab  : $rab (
;  737        P 0831  1                           rac = seq,
;  738        P 0832  1                           rbf = disbuf,
;  739        P 0833  1                           fab = outfab
;  740          0834  1                           ),
;  741          0835  1
;  742          0836  1  !
;  743          0837  1  !        Flag signaling to WILD_USER that a match was found.  This flag
;  744          0838  1  !        is set by the action routine called by WILD_USER.
;  745          0839  1  !
;  746          0840  1          found_match,                   ! found at least one wildcard match
;  747          0841  1
;  748          0842  1  !
;  749          0843  1  ! Wildcard parsing flags set by PARSE_WILD for use in WILD_USER.
;  750          0844  1  !
;  751          0845  1          uic_flag        : long,
```

```
;  752          0846  1          grp_wild        : long,
;  753          0847  1          mem_wild        : long,
;  754          0848  1          str_wild        : long;
;  755          0849  1
;  756          0850  1 global bind
;  757          0851  1          tokenlen        = tokendsc [dsc$w_length] : word,
;  758          0852  1          tokenptr        = tokendsc [dsc$a_pointer],
;  759          0853  1          rec_user_dsc    = uplit (uaf$s_username, recbuf [uaf$t_username]),
;  760          0854  1          rec_encrypt_dsc = uplit (uaf$s_pwd, recbuf [uaf$q_pwd]),
;  761          0855  1          symbol_str      = cstring ('ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789$_');
;  762          0856  1                                    ! valid username characters
;  763          0857  1
;  764          0858  1 !
;  765          0859  1 !       Prompt strings
;  766          0860  1 !
;  767          0861  1
;  768          0862  1 bind
;  769          0863  1
;  770          0864  1     accprmpt     = cstring (%char (lf),'UAF> '),
;  771          0865  1     accprmpt2    = cstring (%char (lf),'_ '),
;  772          0866  1     newmsg20     = cstring ('Do you want to create a new file? ');
;  773          0867  1
;  774          0868  1 !
;  775          0869  1 !       External messages
;  776          0870  1 !
;  777          0871  1
;  778          0872  1 external routine
;  779          0873  1
;  780          0874  1     LIB$SIGNAL;
;  781          0875  1
;  782          0876  1 external literal
;  783          0877  1
;  784          0878  1     UAF$_ADDERR,          UAF$_ADDMSG,          UAF$_BADNODFORM,
;  785          0879  1     UAF$_BADSPC,          UAF$_BADUSR,          UAF$_CLIWARNMSG,
;  786          0880  1     UAF$_CMDTOOLONG,      UAF$_CONERR,          UAF$_COPMSG,
;  787          0881  1     UAF$_CREERR,          UAF$_DEFERR,          UAF$_DEFPWD,
;  788          0882  1     UAF$_DONEMSG,         UAF$_GETERR,          UAF$_HELPERR,
;  789          0883  1     UAF$_INVCMD,          UAF$_INVRSP,          UAF$_INVUSERNAME,
;  790          0884  1     UAF$_KEYNOTFND,       UAF$_KEYNOTUNQ,       UAF$_LSTERR,
;  791          0885  1     UAF$_LSTMSG1,         UAF$_LSTMSG2,         UAF$_MDFYERR,
;  792          0886  1     UAF$_MDFYMSG,         UAF$_NAFADDERR,       UAF$_NAFADDMSG,
;  793          0887  1     UAF$_NAFAEX,          UAF$_NAFCONERR,       UAF$_NAFCREERR,
;  794          0888  1     UAF$_NAFDNE,          UAF$_NAFDONEMSG,      UAF$_NAFNOMODS,
;  795          0889  1     UAF$_NAFUAEERR,       UAF$_NAMETOOBIG,      UAF$_NETLSTMSG,
;  796          0890  1     UAF$_NAOFIL,          UAF$_NAONAF,          UAF$_NOARG,
;  797          0891  1     UAF$_NODEFPWD,        UAF$_NODTOOBIG,       UAF$_NOMODS,
;  798          0892  1     UAF$_NOTUNQ,          UAF$_NOUSERNAME,      UAF$_PREMMSG,
;  799          0893  1     UAF$_PUTERR,          UAF$_RDBDONEMSG,      UAF$_RDBMDFYERR,
;  800          0894  1     UAF$_RDBMDFYERRU,     UAF$_RDBMDFYMSG,      UAF$_RDBNOMODS,
;  801          0895  1     UAF$_REMDEF,          UAF$_REMERR,          UAF$_REMMSG,
;  802          0896  1     UAF$_REMSYS,          UAF$_RENDEF,          UAF$_RENMSG,
;  803          0897  1     UAF$_RENSYS,          UAF$_RONLY,           UAF$_SHOW_ERR,
;  804          0898  1     UAF$_SYSMSG1,         UAF$_SYSMSG2,         UAF$_UAEERR,
;  805          0899  1     UAF$_UICERR,          UAF$_ZISQUAL,         UAF$_ZZPRACREN;
;  806          0900  1
```

```
;  808     0901  1 %sbttl 'start - controlling code'
;  809     0902  1 routine start =
;  810     0903  2 begin
;  811     0904  2
;  812     0905  2 !++
;  813     0906  2 !
;  814     0907  2 ! FUNCTIONAL DESCRIPTION:
;  815     0908  2 !
;  816     0909  2 !       Main procedure of AUTHORIZE.  Call SETUP to initialize
;  817     0910  2 !       all needed files.  Prompt the user for the functions which
;  818     0911  2 !       he/she wants, and call the proper function service routine.
;  819     0912  2 !
;  820     0913  2 ! INPUTS:
;  821     0914  2 !
;  822     0915  2 !       none
;  823     0916  2 !
;  824     0917  2 ! IMPLICIT INPUTS:
;  825     0918  2 !
;  826     0919  2 !       none
;  827     0920  2 !
;  828     0921  2 ! OUTPUTS:
;  829     0922  2 !
;  830     0923  2 !       None
;  831     0924  2 !
;  832     0925  2 ! IMPLICIT OUTPUTS:
;  833     0926  2 !
;  834     0927  2 !       none
;  835     0928  2 !
;  836     0929  2 ! ROUTINE VALUE:
;  837     0930  2 !
;  838     0931  2 !       none
;  839     0932  2 !
;  840     0933  2 ! SIDE EFFECTS:
;  841     0934  2 !
;  842     0935  2 !       none
;  843     0936  2 !--
;  844     0937  2
;  845     0938  2 own
;  846     0939  2         status;
;  847     0940  2
;  848     0941  2 map
;  849     0942  2         cmdlindsc       : vector;
;  850     0943  2
;  851     0944  2 bind
;  852     0945  2         foreign_cmdlindsc = uplit (cmdbuflen, cmdbuf);
;  853     0946  2
;  854     0947  2 rightslist_modified = false;
;  855     0948  2 netuaf_modified = false;
;  856     0949  2 modify_flag = false;                                    ! note no modifications
;  857     0950  2
;  858     0951  2 !
;  859     0952  2 ! Set up terminal I/O
;  860     0953  2 !
;  861     0954  2
;  862     0955  3 if rmsbad (status = $open (fab = infab))
;  863     0956  2 then
;  864     0957  2     signal_stop (.status);
```

```
;  865     0958  2
;  866     0959  3 if rmsbad (status = $connect (rab = inrab))
;  867     0960  2 then
;  868     0961  2     signal_stop (.status);
;  869     0962  2
;  870     0963  2 $create (fab = outfab);
;  871     0964  2 $connect (rab = outrab);
;  872     0965  2
;  873     0966  2 setup ();
;  874     0967  2
;  875     0968  2 !
;  876     0969  2 ! Files have been initialized.  Prompt user for command line
;  877     0970  2 ! and perform requested function.
;  878     0971  2 !
;  879     0972  2
;  880     0973  2 if lib$get_foreign (foreign_cmdlindsc, 0, cmdlindsc)  and  .cmdlindsc [0] neq 0
;  881     0974  2 then
;  882     0975  2     !
;  883     0976  2     ! If defined foreign, and there are commands on the line...
;  884     0977  2     !
;  885     0978  3     begin
;  886     0979  3     cmdlindsc [1] = cmdbuf;
;  887     0980  3
;  888     0981  4          if (status = cli$dcl_parse (cmdlindsc, authorize_commands))
;  889     0982  3          then
;  890     0983  4             begin
;  891     0984  4             cli$dispatch ();
;  892     0985  4             return true;
;  893     0986  4             end
;  894     0987  3          else
;  895     0988  3               !
;  896     0989  3               ! See if no CLINT exists (Kludge City, USA  37916)
;  897     0990  3               !
;  898     0991  3               if .status eql cli$_noclint
;  899     0992  3               then
;  900     0993  3                   signal_stop (cli$_noclint)
;  901     0994  3               else
;  902     0995  3                   acc$exit ()
;  903     0996  2     end;
;  904     0997  2
;  905     0998  2 while true
;  906     0999  2 do
;  907     1000  3     begin
;  908     1001  3
;  909     1002  3     !
;  910     1003  3     ! Input the command line, taking care of continuations.  Pull off
;  911     1004  3     ! the first token, assuming it is the command name, and look it up
;  912     1005  3     ! in the table of commands.
;  913     1006  3     !
;  914     1007  3
;  915     1008  3     if get_cmd_line ()
;  916     1009  3     then
;  917     1010  4         if (status = cli$dcl_parse (cmdlindsc, authorize_commands))
;  918     1011  3         then
;  919     1012  4             begin
;  920     1013  4             ch$fill (0, uaf$s_flags, uaf$gl_ctlmsk);
;  921     1014  4             cli$dispatch ();
```

```
;  922          1015  4              end
;  923          1016  3          else
;  924          1017  3              if .status eql cli$_noclint
;  925          1018  3              then
;  926          1019  3                  signal_stop (cli$_noclint)
;  927          1020  2      end;
;  928          1021  2
;  929          1022  2 return true;
;  930          1023  2
;  931          1024  1 end;
```

```
                                                                .TITLE  UAFMAIN
                                                                .IDENT  \V04-000\

                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                31  6E  65  6B  6F  74  00000 P.AAB:   .ASCII  \token1\                    ;
                                                    00006           .BLKB   2
                                      00000006      00008 P.AAA:    .LONG   6                           ;
                                      00000000'     0000C           .ADDRESS P.AAB                      ;
                                32  6E  65  6B  6F  74  00010 P.AAD:   .ASCII  \token2\                   ;
                                                    00016           .BLKB   2
                                      00000006      00018 P.AAC:    .LONG   6                           ;
                                      00000000'     0001C           .ADDRESS P.AAD                      ;
                                    66  65  69  72  62  00020 P.AAF:   .ASCII  \brief\                  ;
                                                    00025           .BLKB   3
                                      00000005      00028 P.AAE:    .LONG   5                           ;
                                      00000000'     0002C           .ADDRESS P.AAF                      ;
                                    6C  6C  75  66  00030 P.AAH:    .ASCII  \full\                      ;
                                      00000004      00034 P.AAG:    .LONG   4                           ;
                                      00000000'     00038           .ADDRESS P.AAH                      ;
    72  65  69  66  69  74  6E  65  64  69  5F  64  64  61  0003C P.AAJ:    .ASCII  \add_identifier\            ;
                                                    0004A           .BLKB   2
                                      0000000E      0004C P.AAI:    .LONG   14                          ;
                                      00000000'     00050           .ADDRESS P.AAJ                      ;
69  66  69  74  6E  65  64  69  5F  65  76  6F  6D  65  72  00054 P.AAL:    .ASCII  \remove_identifier\         ;
                                                    00063
                                                    00065           .BLKB   3
                                      00000011      00068 P.AAK:    .LONG   17                          ;
                                      00000000'     0006C           .ADDRESS P.AAL                      ;
69  66  69  74  6E  65  64  69  5F  79  66  69  64  6F  6D  00070 P.AAN:    .ASCII  \modify_identifier\         ;
                                                    0007F
                                                    00081           .BLKB   3
                                      00000011      00084 P.AAM:    .LONG   17                          ;
                                      00000000'     00088           .ADDRESS P.AAN                      ;
6F  73  65  72  2E  73  65  74  75  62  69  72  74  74  61  0008C P.AAO:    .ASCII  \attributes.resource\       ;
                                                    00098
                                                    0009B
                                    46  41  55  53  59  53  0009F P.AAP:    .ASCII  \SYSUAF\                    ;
                                    46  41  55  54  45  4E  000A5 P.AAQ:    .ASCII  \NETUAF\                    ;
                                64  65  69  66  69  64  6F  6D  000AB P.AAS:    .ASCII  \modified\                  ;
                                                    000B3           .BLKB   1
                                      00000008      000B4 P.AAR:    .LONG   8                           ;
                                      00000000'     000B8           .ADDRESS P.AAS                      ;
                                        64  65  64  64  61  000BC P.AAU:    .ASCII  \added\                     ;
                                                    000C1           .BLKB   3
                                      00000005      000C4 P.AAT:    .LONG   5                           ;
```

```
                                                00000000'  000C8          .ADDRESS P.AAU                                  ;
                                        64 65 76 6F 6D 65 72  000CC P.AAW:  .ASCII  \removed\                               ;
                                                              000D3         .BLKB   1                                       ;
                                                00000007      000D4 P.AAV:  .LONG   7                                       ;
                                                00000000'     000D8         .ADDRESS P.AAW                                  ;
53 41 21 20 53 41 21 20 4C 58 21 3D 44 49 50  000DC P.AAY:  .ASCII  \PID=!XL !AS !AS !AS record !AS on !%D\                ;
53 41 21 20 64 72 6F 63 65 72 20 53 41 21 20  000EB                                                                        ;
                        44 25 21 20 6E 6F 20  000FA                                                                        ;
                                              00101         .BLKB   3                                                      ;
                                                00000025      00104 P.AAX:  .LONG   37                                      ;
                                                00000000'     00108         .ADDRESS P.AAY                                  ;
                                                   3A 3A      0010C P.ABA:  .ASCII  \::\                                    ;
                                                              0010E         .BLKB   2                                       ;
                                                00000002      00110 P.AAZ:  .LONG   2                                       ;
                                                00000000'     00114         .ADDRESS P.ABA                                  ;
                                     FFFFFFFF   FFB3B4C0      00118 P.ABB:  .LONG   -5000000, -1                            ;
                                                      07      00120 P.ABC:  .BYTE   7                                       ;
                                        54 4C 55 41 46 45 44  00121         .ASCII  \DEFAULT\                               ;
                                                      04      00128 P.ABD:  .BYTE   4                                       ;
                                              52 45 53 55     00129         .ASCII  \USER\                                  ;
                                                      09      0012D P.ABE:  .BYTE   9                                       ;
                              53 45 4C 42 41 54 4C 43 44      0012E         .ASCII  \DCLTABLES\                             ;
                                                      00      00137 P.ABF:  .BYTE   0                                       ;
                                                              00138         .BLKB   0                                       ;
                                                      03      00138 P.ABG:  .BYTE   3                                       ;
                                                 4C 43 44     00139         .ASCII  \DCL\                                   ;
                                                      00      0013C P.ABH:  .BYTE   0                                       ;
                                                              0013D         .BLKB   0                                       ;
                                                      00      0013D P.ABI:  .BYTE   0                                       ;
                                                              0013E         .BLKB   0                                       ;
                                                      06      0013E P.ABJ:  .BYTE   6                                       ;
                                        5D 52 45 53 55 5B     0013F         .ASCII  \[USER]\                                ;
                                                      00      00145 P.ABK:  .BYTE   0                                       ;
                                                              00146         .BLKB   0                                       ;
                                                              00146         .BLKB   2                                       ;
                                     00000000   00108000      00148 P.ABL:  .LONG   1081344, 0                              ;
                                     00000000   00000000      00150 P.ABM:  .LONG   0, 0                                    ;
                                                      06      00158 P.ABN:  .BYTE   6                                       ;
                                           4D 45 54 53 59 53  00159         .ASCII  \SYSTEM\                                ;
                                                      07      0015F P.ABO:  .BYTE   7                                       ;
                                     52 45 47 41 4E 41 4D     00160         .ASCII  \MANAGER\                               ;
                                                      09      00167 P.ABP:  .BYTE   9                                       ;
                              53 45 4C 42 41 54 4C 43 44      00168         .ASCII  \DCLTABLES\                             ;
                                                      06      00171 P.ABQ:  .BYTE   6                                       ;
                                           4D 45 54 53 59 53  00172         .ASCII  \SYSTEM\                                ;
                                                      03      00178 P.ABR:  .BYTE   3                                       ;
                                                 4C 43 44     00179         .ASCII  \DCL\                                   ;
                                                      0E      0017C P.ABS:  .BYTE   14                                      ;
52 45 47 41 4E 41 4D 20 4D 45 54 53 59 53  0017D         .ASCII  \SYSTEM MANAGER\                                        ;
                                                      00      0018B P.ABT:  .BYTE   0                                       ;
                                                              0018C         .BLKB   0                                       ;
                                                      08      0018C P.ABU:  .BYTE   8                                       ;
                                 5D 52 47 4D 53 59 53 5B      0018D         .ASCII  \[SYSMGR]\                              ;
                                                      00      00195 P.ABV:  .BYTE   0                                       ;
                                                              00196         .BLKB   0                                       ;
                                                              00196         .BLKB   2                                       ;
                                                FFFFFFFF#     00198 P.ABW:  .LONG   -1[2]                                   ;
```

```
                                                00000000  00000000  001A0 P.ABX:   .LONG   0, 0                                    ;
                                    54  55  50  4E  49  24  53  59  53  001A8 P.ABY:   .ASCII  \SYS$INPUT\                          ;
                                54  55  50  54  55  4F  24  53  59  53  001B1 P.ABZ:   .ASCII  \SYS$OUTPUT\                         ;
                                53  49  4C  2E  46  41  55  53  59  53  001BB P.ACA:   .ASCII  \SYSUAF.LIS\                        ;
                                53  49  4C  2E  46  41  55  54  45  4E  001C5 P.ACB:   .ASCII  \NETUAF.LIS\                       ;
                                                46  41  55  53  59  53  001CF P.ACC:   .ASCII  \SYSUAF\                           ;
                                                        54  41  44  2E  001D5 P.ACD:   .ASCII  \.DAT\                            ;
                                                46  41  55  54  45  4E  001D9 P.ACE:   .ASCII  \NETUAF\                          ;
                                                        54  41  44  2E  001DF P.ACF:   .ASCII  \.DAT\                            ;
                                                                        001E3          .BLKB   1                                 ;
                                                            00000020    001E4 P.ACG:   .LONG   32                                ;
                                                            00000000'   001E8          .ADDRESS RECBUF+4                         ;
                                                            00000008    001EC P.ACH:   .LONG   8                                 ;
                                                            00000000'   001F0          .ADDRESS RECBUF+340                       ;
                                                                    26  001F4 P.ACI:   .BYTE   38                                ;
4F  4E  4D  4C  4B  4A  49  48  47  46  45  44  43  42  41           001F5          .ASCII  \ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789$_\  ;
33  32  31  30  5A  59  58  57  56  55  54  53  52  51  50           00204                                                            ;
                        5F  24  39  38  37  36  35  34               00213                                                            ;
                                                    06  0021B P.ACJ:   .BYTE   6                                               ;
                                                20  3E  46  41  55  0A  0021C          .ASCII  <10>\UAF> \                       ;
                                                                    03  00222 P.ACK:   .BYTE   3                                ;
                                                            20  5F  0A  00223          .ASCII  <10>\_ \                          ;
                                                                    22  00226 P.ACL:   .BYTE   34                               ;
20  6F  74  20  74  6E  61  77  20  75  6F  79  20  6F  44           00227          .ASCII  \Do you want to create a new file? \      ;
69  66  20  77  65  6E  20  61  20  65  74  61  65  72  63           00236                                                            ;
                                                20  3F  65  6C  00245                                                            ;
                                                                    00249          .BLKB   3                                 ;
                                                            00000400    0024C P.ACM:   .LONG   1024                              ;
                                                            00000000'   00250          .ADDRESS CMDBUF                           ;

                                                                                   .PSECT  $OWN$,NOEXE,2

                                                0013  00000 SD_ATTRIBRESOURCE:
                                                                    .WORD   19                                                 ;
                                              01  0E  00002          .BYTE   14, 1                                             ;
                                            00000000'   00004          .ADDRESS P.AAO                                        ;
                                                        00008 USERNAME_BUF:
                                                                    .BLKB   32
                                                        00028 PCB_STS:.BLKB   4
                                                        0002C PID:    .BLKB   4
                                            00000000    00030 USERNAME_DSC:
                                                                    .LONG   0                                                  ;
                                            00000000'   00034          .ADDRESS USERNAME_BUF                                 ;
                                                        00038 RECNAME_DSC:
                                                                    .BLKB   8
                                            00000006    00040 FILE_DSC:
                                                                    .LONG   6                                                  ;
                                                        00044          .BLKB   4
                                                        00048 MOD_DEFAULT:
                                                                    .BLKB   4
                                                        0004C MODIFY_FLAG:
                                                                    .BLKB   4
                                                        00050 NETUAF_MODIFIED:
                                                                    .BLKB   4
                                                  00    00054 RENAME_PH2:
                                                                    .BYTE   0                                                  ;
                                                        00055          .BLKB   3
```

```
                        00058 OLDUSERLEN:
                                      .BLKB   4
                        0005C OLDUSERNAME:
                                      .BLKB   32
                        0007C NEWUSERLEN:
                                      .BLKB   4
                        00080 NEWUSERNAME:
                                      .BLKB   32
                        000A0 CMDBUF: .BLKB   1024
                        004A0 DEFAULT_SIZE:
                                      .BLKB   2
                        004A2         .BLKB   2
                        004A4 DEFAULT_RECORD:
                                      .BLKB   1412
                        00A28 PWDDSC: .BLKB   8
                        00A30 INSIZE: .BLKB   4
                        00A34 BRIEF_FLAG:
                                      .BLKB   4
                        00A38 FULL_FLAG:
                                      .BLKB   4
                        00A3C HEADER_FLAG:
                                      .BLKB   4
              03        00A40 INFAB:  .BYTE   3
              50        00A41         .BYTE   80
            0000        00A42         .WORD   0
        00000000        00A44         .LONG   0
        00000000        00A48         .LONG   0
        00000000        00A4C         .LONG   0
        00000000        00A50         .LONG   0
            0000        00A54         .WORD   0
              03        00A56         .BYTE   3
              00        00A57         .BYTE   0
        00000000        00A58         .LONG   0
              00        00A5C         .BYTE   0
              00        00A5D         .BYTE   0
              02        00A5E         .BYTE   2
              02        00A5F         .BYTE   2
        00000000        00A60         .LONG   0
        00000000        00A64         .LONG   0
        00000000        00A68         .LONG   0
        00000000'       00A6C         .ADDRESS P.ABY
        00000000        00A70         .LONG   0
              09        00A74         .BYTE   9
              00        00A75         .BYTE   0
            0000        00A76         .WORD   0
        00000000        00A78         .LONG   0
            0000        00A7C         .WORD   0
              00        00A7E         .BYTE   0
              00        00A7F         .BYTE   0
        00000000        00A80         .LONG   0
        00000000        00A84         .LONG   0
            0000        00A88         .WORD   0
              00        00A8A         .BYTE   0
              00        00A8B         .BYTE   0
        00000000        00A8C         .LONG   0
              01        00A90 INRAB:  .BYTE   1
              44        00A91         .BYTE   68
```

```
                0000  00A92                .WORD   0
            40000000  00A94                .LONG   1073741824
            00000000  00A98                .LONG   0
            00000000  00A9C                .LONG   0
                0000# 00AA0                .WORD   0[3]
                0000  00AA6                .WORD   0
            00000000  00AA8                .LONG   0
                0000  00AAC                .WORD   0
                  00  00AAE                .BYTE   0
                  00  00AAF                .BYTE   0
                0000  00AB0                .WORD   0
                0000  00AB2                .WORD   0
            00000000  00AB4                .LONG   0
            00000000  00AB8                .LONG   0
            00000000  00ABC                .LONG   0
            00000000  00AC0                .LONG   0
                  00  00AC4                .BYTE   0
                  00  00AC5                .BYTE   0
                  00  00AC6                .BYTE   0
                  00  00AC7                .BYTE   0
            00000000  00AC8                .LONG   0
            00000000' 00ACC                .ADDRESS INFAB
            00000000  00AD0                .LONG   0
                  03  00AD4 OUTFAB:        .BYTE   3
                  50  00AD5                .BYTE   80
                0000  00AD6                .WORD   0
            00000000  00AD8                .LONG   0
            00000000  00ADC                .LONG   0
            00000000  00AE0                .LONG   0
            00000000  00AE4                .LONG   0
                0000  00AE8                .WORD   0
                  01  00AEA                .BYTE   1
                  00  00AEB                .BYTE   0
            00000000  00AEC                .LONG   0
                  00  00AF0                .BYTE   0
                  00  00AF1                .BYTE   0
                  02  00AF2                .BYTE   2
                  02  00AF3                .BYTE   2
            00000000  00AF4                .LONG   0
            00000000  00AF8                .LONG   0
            00000000  00AFC                .LONG   0
            00000000' 00B00                .ADDRESS P.ABZ
            00000000  00B04                .LONG   0
                  0A  00B08                .BYTE   10
                  00  00B09                .BYTE   0
                0000  00B0A                .WORD   0
            00000000  00B0C                .LONG   0
                0000  00B10                .WORD   0
                  00  00B12                .BYTE   0
                  00  00B13                .BYTE   0
            00000000  00B14                .LONG   0
            00000000  00B18                .LONG   0
                0000  00B1C                .WORD   0
                  00  00B1E                .BYTE   0
                  00  00B1F                .BYTE   0
            00000000  00B20                .LONG   0
                  02  00B24 LSTNAM:        .BYTE   2
```

```
      60  00B25          .BYTE   96
      00  00B26          .BYTE   0
      00  00B27          .BYTE   0
00000000  00B28          .LONG   0
      00  00B2C          .BYTE   0
      00  00B2D          .BYTE   0
      00  00B2E          .BYTE   0
      00  00B2F          .BYTE   0
00000000  00B30          .LONG   0
00000000  00B34          .LONG   0
    0000# 00B38          .WORD   0[8]
    0000# 00B48          .WORD   0[3]
    0000# 00B4E          .WORD   0[3]
00000000  00B54          .LONG   0
00000000  00B58          .LONG   0
      00  00B5C          .BYTE   0
      00  00B5D          .BYTE   0
      00  00B5E          .BYTE   0
      00  00B5F          .BYTE   0
      00  00B60          .BYTE   0
      00  00B61          .BYTE   0
      00# 00B62          .BYTE   0[2]
00000000  00B64          .LONG   0
00000000  00B68          .LONG   0
00000000  00B6C          .LONG   0
00000000  00B70          .LONG   0
00000000  00B74          .LONG   0
00000000  00B78          .LONG   0
00000000# 00B7C          .LONG   0[2]
      13  00B84 LSTPRO:  .BYTE   19
      58  00B85          .BYTE   88
    0000  00B86          .WORD   0
00000000  00B88          .LONG   0
    FC44  00B8C          .WORD   -956
      00  00B8E          .BYTE   0
      00  00B8F          .BYTE   0
0000  0000  00B90        .WORD   0, 0
      00  00B94          .BYTE   0
      00  00B95          .BYTE   0
    0000  00B96          .WORD   0
00000000  00B98          .LONG   0
00000000  00B9C          .LONG   0
    0000  00BA0          .WORD   0
    0000  00BA2          .WORD   0
00000000  00BA4          .LONG   0
00000000  00BA8          .LONG   0
          00BAC          .BLKB   48
      03  00BDC LSTFAB:  .BYTE   3
      50  00BDD          .BYTE   80
    0000  00BDE          .WORD   0
00000000  00BE0          .LONG   0
00000000  00BE4          .LONG   0
00000000  00BE8          .LONG   0
00000000  00BEC          .LONG   0
    0000  00BF0          .WORD   0
      01  00BF2          .BYTE   1
      20  00BF3          .BYTE   32
```

```
00000000  00BF4           .LONG   0
       00  00BF8           .BYTE   0
       00  00BF9           .BYTE   0
       02  00BFA           .BYTE   2
       02  00BFB           .BYTE   2
00000000  00BFC           .LONG   0
00000000' 00C00           .ADDRESS LSTPRO
00000000' 00C04           .ADDRESS LSTNAM
00000000' 00C08           .ADDRESS P.ACA
00000000  00C0C           .LONG   0
       0A  00C10           .BYTE   10
       00  00C11           .BYTE   0
     0084  00C12           .WORD   132
00000000  00C14           .LONG   0
     0000  00C18           .WORD   0
       00  00C1A           .BYTE   0
       00  00C1B           .BYTE   0
00000000  00C1C           .LONG   0
00000000  00C20           .LONG   0
     0000  00C24           .WORD   0
       00  00C26           .BYTE   0
       00  00C27           .BYTE   0
00000000  00C28           .LONG   0
       01  00C2C LSTRAB:   .BYTE   1
       44  00C2D           .BYTE   68
     0000  00C2E           .WORD   0
00000000  00C30           .LONG   0
00000000  00C34           .LONG   0
00000000  00C38           .LONG   0
     0000# 00C3C           .WORD   0[3]
     0000  00C42           .WORD   0
00000000  00C44           .LONG   0
     0000  00C48           .WORD   0
       00  00C4A           .BYTE   0
       00  00C4B           .BYTE   0
     0000  00C4C           .WORD   0
     0000  00C4E           .WORD   0
00000000  00C50           .LONG   0
00000000' 00C54           .ADDRESS DISBUF
00000000  00C58           .LONG   0
00000000  00C5C           .LONG   0
       00  00C60           .BYTE   0
       00  00C61           .BYTE   0
       00  00C62           .BYTE   0
       00  00C63           .BYTE   0
00000000  00C64           .LONG   0
00000000' 00C68           .ADDRESS LSTFAB
00000000  00C6C           .LONG   0
       02  00C70 NLSTNAM:.BYTE   2
       60  00C71           .BYTE   96
       00  00C72           .BYTE   0
       00  00C73           .BYTE   0
00000000  00C74           .LONG   0
       00  00C78           .BYTE   0
       00  00C79           .BYTE   0
       00  00C7A           .BYTE   0
       00  00C7B           .BYTE   0
```

```
00000000  00C7C            .LONG   0
00000000  00C80            .LONG   0
    0000# 00C84            .WORD   0[8]
    0000# 00C94            .WORD   0[3]
    0000# 00C9A            .WORD   0[3]
00000000  00CA0            .LONG   0
00000000  00CA4            .LONG   0
       00  00CA8            .BYTE   0
       00  00CA9            .BYTE   0
       00  00CAA            .BYTE   0
       00  00CAB            .BYTE   0
       00  00CAC            .BYTE   0
       00  00CAD            .BYTE   0
       00# 00CAE            .BYTE   0[2]
00000000  00CB0            .LONG   0
00000000  00CB4            .LONG   0
00000000  00CB8            .LONG   0
00000000  00CBC            .LONG   0
00000000  00CC0            .LONG   0
00000000  00CC4            .LONG   0
00000000# 00CC8            .LONG   0[2]
       13  00CD0 NLSTPRO:  .BYTE   19
       58  00CD1            .BYTE   88
     0000  00CD2            .WORD   0
00000000  00CD4            .LONG   0
     FC44  00CD8            .WORD   -956
       00  00CDA            .BYTE   0
       00  00CDB            .BYTE   0
0000 0000  00CDC            .WORD   0, 0
       00  00CE0            .BYTE   0
       00  00CE1            .BYTE   0
     0000  00CE2            .WORD   0
00000000  00CE4            .LONG   0
00000000  00CE8            .LONG   0
     0000  00CEC            .WORD   0
     0000  00CEE            .WORD   0
00000000  00CF0            .LONG   0
00000000  00CF4            .LONG   0
           00CF8            .BLKB   48
       03  00D28 NLSTFAB:  .BYTE   3
       50  00D29            .BYTE   80
     0000  00D2A            .WORD   0
00000000  00D2C            .LONG   0
00000000  00D30            .LONG   0
00000000  00D34            .LONG   0
00000000  00D38            .LONG   0
     0000  00D3C            .WORD   0
       01  00D3E            .BYTE   1
       20  00D3F            .BYTE   32
00000000  00D40            .LONG   0
       00  00D44            .BYTE   0
       00  00D45            .BYTE   0
       02  00D46            .BYTE   2
       02  00D47            .BYTE   2
00000000  00D48            .LONG   0
00000000' 00D4C            .ADDRESS NLSTPRO
00000000' 00D50            .ADDRESS NLSTNAM
```

```
00000000'  00D54                .ADDRESS P.ACB                                  ;
00000000   00D58                .LONG    0                                      ;
        0A 00D5C                .BYTE    10                                     ;
        00 00D5D                .BYTE    0                                      ;
      0084 00D5E                .WORD    132                                    ;
00000000   00D60                .LONG    0                                      ;
      0000 00D64                .WORD    0                                      ;
        00 00D66                .BYTE    0                                      ;
        00 00D67                .BYTE    0                                      ;
00000000   00D68                .LONG    0                                      ;
00000000   00D6C                .LONG    0                                      ;
      0000 00D70                .WORD    0                                      ;
        00 00D72                .BYTE    0                                      ;
        00 00D73                .BYTE    0                                      ;
00000000   00D74                .LONG    0                                      ;
        01 00D78 NLSTRAB:       .BYTE    1                                      ;
        44 00D79                .BYTE    68                                     ;
      0000 00D7A                .WORD    0                                      ;
00000000   00D7C                .LONG    0                                      ;
00000000   00D80                .LONG    0                                      ;
00000000   00D84                .LONG    0                                      ;
     0000# 00D88                .WORD    0[3]                                   ;
      0000 00D8E                .WORD    0                                      ;
00000000   00D90                .LONG    0                                      ;
      0000 00D94                .WORD    0                                      ;
        00 00D96                .BYTE    0                                      ;
        00 00D97                .BYTE    0                                      ;
      0000 00D98                .WORD    0                                      ;
      0000 00D9A                .WORD    0                                      ;
00000000   00D9C                .LONG    0                                      ;
00000000'  00DA0                .ADDRESS DISBUF                                 ;
00000000   00DA4                .LONG    0                                      ;
00000000   00DA8                .LONG    0                                      ;
        00 00DAC                .BYTE    0                                      ;
        00 00DAD                .BYTE    0                                      ;
        00 00DAE                .BYTE    0                                      ;
        00 00DAF                .BYTE    0                                      ;
00000000   00DB0                .LONG    0                                      ;
00000000'  00DB4                .ADDRESS NLSTFAB                                ;
00000000   00DB8                .LONG    0                                      ;
        15 00DBC UAFKEY2:       .BYTE    21                                     ;
        4C 00DBD                .BYTE    76                                     ;
      0000 00DBE                .WORD    0                                      ;
00000000   00DC0                .LONG    0                                      ;
        00 00DC4                .BYTE    0                                      ;
        00 00DC5                .BYTE    0                                      ;
        00 00DC6                .BYTE    0                                      ;
        00 00DC7                .BYTE    0                                      ;
        00 00DC8                .BYTE    0                                      ;
        00 00DC9                .BYTE    0                                      ;
00000000   00DCA                .LONG    0                                      ;
        03 00DCE                .BYTE    3                                      ;
        02 00DCF                .BYTE    2                                      ;
        00 00DD0                .BYTE    0                                      ;
        00 00DD1                .BYTE    0                                      ;
        00 00DD2                .BYTE    0                                      ;
        02 00DD3                .BYTE    2                                      ;
```

```
                0000  00DD4                .WORD   0
                0000  00DD6                .WORD   0
                0000  00DD8                .WORD   0
                0024  00DDA                .WORD   36
                0000  00DDC                .WORD   0
                0000  00DDE                .WORD   0
                0000  00DE0                .WORD   0
                0000  00DE2                .WORD   0
                0000  00DE4                .WORD   0
                0000  00DE6                .WORD   0
                0000  00DE8                .WORD   0
                  02  00DEA                .BYTE   2
                  00  00DEB                .BYTE   0
                  00  00DEC                .BYTE   0
                  00  00DED                .BYTE   0
                  00  00DEE                .BYTE   0
                  00  00DEF                .BYTE   0
                  00  00DF0                .BYTE   0
                  00  00DF1                .BYTE   0
                0000  00DF2                .WORD   0
            00000000  00DF4                .LONG   0
            00000000  00DF8                .LONG   0
                  00  00DFC                .BYTE   0
                  00  00DFD                .BYTE   0
                  00  00DFE                .BYTE   0
                  00  00DFF                .BYTE   0
                  00  00E00                .BYTE   0
                  00  00E01                .BYTE   0
                  00  00E02                .BYTE   0
                  00  00E03                .BYTE   0
                  00  00E04                .BYTE   0
                  00  00E05                .BYTE   0
                      00E06                .BLKB   2
                  15  00E08  UAFKEY1:      .BYTE   21
                  4C  00E09                .BYTE   76
                0000  00E0A                .WORD   0
           00000000'  00E0C                .ADDRESS UAFKEY2
                  00  00E10                .BYTE   0
                  00  00E11                .BYTE   0
                  00  00E12                .BYTE   0
                  00  00E13                .BYTE   0
                  00  00E14                .BYTE   0
                  00  00E15                .BYTE   0
            00000000  00E16                .LONG   0
                  03  00E1A                .BYTE   3
                  04  00E1B                .BYTE   4
                  00  00E1C                .BYTE   0
                  00  00E1D                .BYTE   0
                  00  00E1E                .BYTE   0
                  01  00E1F                .BYTE   1
                0000  00E20                .WORD   0
                0000  00E22                .WORD   0
                0000  00E24                .WORD   0
                0024  00E26                .WORD   36
                0000  00E28                .WORD   0
                0000  00E2A                .WORD   0
                0000  00E2C                .WORD   0
```

```
                0000  00E2E            .WORD   0
                0000  00E30            .WORD   0
                0000  00E32            .WORD   0
                0000  00E34            .WORD   0
                  04  00E36            .BYTE   4
                  00  00E37            .BYTE   0
                  00  00E38            .BYTE   0
                  00  00E39            .BYTE   0
                  00  00E3A            .BYTE   0
                  00  00E3B            .BYTE   0
                  00  00E3C            .BYTE   0
                  00  00E3D            .BYTE   0
                0000  00E3E            .WORD   0
            00000000  00E40            .LONG   0
            00000000  00E44            .LONG   0
                  00  00E48            .BYTE   0
                  00  00E49            .BYTE   0
                  00  00E4A            .BYTE   0
                  00  00E4B            .BYTE   0
                  00  00E4C            .BYTE   0
                  00  00E4D            .BYTE   0
                  00  00E4E            .BYTE   0
                  00  00E4F            .BYTE   0
                  00  00E50            .BYTE   0
                  00  00E51            .BYTE   0
                      00E52            .BLKB   2
                  15  00E54 UAFKEY0:   .BYTE   21
                  4C  00E55            .BYTE   76
                0000  00E56            .WORD   0
            00000000' 00E58            .ADDRESS UAFKEY1
                  00  00E5C            .BYTE   0
                  00  00E5D            .BYTE   0
                  00  00E5E            .BYTE   0
                  00  00E5F            .BYTE   0
                  00  00E60            .BYTE   0
                  00  00E61            .BYTE   0
            00000000  00E62            .LONG   0
                  00  00E66            .BYTE   0
                  00  00E67            .BYTE   0
                  00  00E68            .BYTE   0
                  00  00E69            .BYTE   0
                  00  00E6A            .BYTE   0
                  00  00E6B            .BYTE   0
                0000  00E6C            .WORD   0
                0000  00E6E            .WORD   0
                0000  00E70            .WORD   0
                0004  00E72            .WORD   4
                0000  00E74            .WORD   0
                0000  00E76            .WORD   0
                0000  00E78            .WORD   0
                0000  00E7A            .WORD   0
                0000  00E7C            .WORD   0
                0000  00E7E            .WORD   0
                0000  00E80            .WORD   0
                  20  00E82            .BYTE   32
                  00  00E83            .BYTE   0
                  00  00E84            .BYTE   0
```

```
                      00  00E85          .BYTE   0
                      00  00E86          .BYTE   0
                      00  00E87          .BYTE   0
                      00  00E88          .BYTE   0
                      00  00E89          .BYTE   0
                    0000  00E8A          .WORD   0
                00000000  00E8C          .LONG   0
                00000000  00E90          .LONG   0
                      00  00E94          .BYTE   0
                      00  00E95          .BYTE   0
                      00  00E96          .BYTE   0
                      00  00E97          .BYTE   0
                      00  00E98          .BYTE   0
                      00  00E99          .BYTE   0
                      00  00E9A          .BYTE   0
                      00  00E9B          .BYTE   0
                      00  00E9C          .BYTE   0
                      00  00E9D          .BYTE   0
                          00E9E          .BLKB   2
                      13  00EA0 UAFPRO:  .BYTE   19
                      58  00EA1          .BYTE   88
                    0000  00EA2          .WORD   0
                00000000' 00EA4          .ADDRESS UAFKEY0
                    FF00  00EA8          .WORD   -256
                      00  00EAA          .BYTE   0
                      00  00EAB          .BYTE   0
              0000  0000  00EAC          .WORD   0, 0
                      00  00EB0          .BYTE   0
                      00  00EB1          .BYTE   0
                    0000  00EB2          .WORD   0
                00000000  00EB4          .LONG   0
                00000000  00EB8          .LONG   0
                    0000  00EBC          .WORD   0
                    0000  00EBE          .WORD   0
                00000000  00EC0          .LONG   0
                00000000  00EC4          .LONG   0
                          00EC8          .BLKB   48
                      03  00EF8 UAFFAB:  .BYTE   3
                      50  00EF9          .BYTE   80
                    0000  00EFA          .WORD   0
                02000000  00EFC          .LONG   33554432
                00000000  00F00          .LONG   0
                00000000  00F04          .LONG   0
                0000000A  00F08          .LONG   10
                    000A  00F0C          .WORD   10
                      0F  00F0E          .BYTE   15
                      0F  00F0F          .BYTE   15
                00000000  00F10          .LONG   0
                      00  00F14          .BYTE   0
                      20  00F15          .BYTE   32
                      00  00F16          .BYTE   0
                      02  00F17          .BYTE   2
                00000000  00F18          .LONG   0
                00000000' 00F1C          .ADDRESS UAFPRO
                00000000  00F20          .LONG   0
                00000000' 00F24          .ADDRESS P.ACC
                00000000' 00F28          .ADDRESS P.ACD
```

```
                06  00F2C          .BYTE   6
                04  00F2D          .BYTE   4
              0584  00F2E          .WORD   1412
        00000000  00F30          .LONG   0
              0000  00F34          .WORD   0
                00  00F36          .BYTE   0
                00  00F37          .BYTE   0
        00000000  00F38          .LONG   0
        00000000  00F3C          .LONG   0
              0000  00F40          .WORD   0
                00  00F42          .BYTE   0
                00  00F43          .BYTE   0
        00000000  00F44          .LONG   0
                15  00F48 NAFKEY1:.BYTE   21
                4C  00F49          .BYTE   76
              0000  00F4A          .WORD   0
        00000000  00F4C          .LONG   0
                00  00F50          .BYTE   0
                00  00F51          .BYTE   0
                00  00F52          .BYTE   0
                00  00F53          .BYTE   0
                00  00F54          .BYTE   0
                00  00F55          .BYTE   0
        00000000  00F56          .LONG   0
                03  00F5A          .BYTE   3
                00  00F5B          .BYTE   0
                00  00F5C          .BYTE   0
                00  00F5D          .BYTE   0
                00  00F5E          .BYTE   0
                01  00F5F          .BYTE   1
              0000  00F60          .WORD   0
              0000  00F62          .WORD   0
              0000  00F64          .WORD   0
              0040  00F66          .WORD   64
              0000  00F68          .WORD   0
              0000  00F6A          .WORD   0
              0000  00F6C          .WORD   0
              0000  00F6E          .WORD   0
              0000  00F70          .WORD   0
              0000  00F72          .WORD   0
              0000  00F74          .WORD   0
                20  00F76          .BYTE   32
                00  00F77          .BYTE   0
                00  00F78          .BYTE   0
                00  00F79          .BYTE   0
                00  00F7A          .BYTE   0
                00  00F7B          .BYTE   0
                00  00F7C          .BYTE   0
                00  00F7D          .BYTE   0
              0000  00F7E          .WORD   0
        00000000  00F80          .LONG   0
        00000000  00F84          .LONG   0
                00  00F88          .BYTE   0
                00  00F89          .BYTE   0
                00  00F8A          .BYTE   0
                00  00F8B          .BYTE   0
                00  00F8C          .BYTE   0
```

```
                00  00F8D           .BYTE   0                                                       ;
                00  00F8E           .BYTE   0                                                       ;
                00  00F8F           .BYTE   0                                                       ;
                00  00F90           .BYTE   0                                                       ;
                00  00F91           .BYTE   0                                                       ;
                    00F92           .BLKB   2                                                       ;
                15  00F94 NAFKEY0:  .BYTE   21                                                      ;
                4C  00F95           .BYTE   76                                                      ;
              0000  00F96           .WORD   0                                                       ;
        00000000'   00F98           .ADDRESS NAFKEY1                                                ;
                00  00F9C           .BYTE   0                                                       ;
                00  00F9D           .BYTE   0                                                       ;
                00  00F9E           .BYTE   0                                                       ;
                00  00F9F           .BYTE   0                                                       ;
                00  00FA0           .BYTE   0                                                       ;
                00  00FA1           .BYTE   0                                                       ;
        00000000    00FA2           .LONG   0                                                       ;
                00  00FA6           .BYTE   0                                                       ;
                00  00FA7           .BYTE   0                                                       ;
                00  00FA8           .BYTE   0                                                       ;
                00  00FA9           .BYTE   0                                                       ;
                00  00FAA           .BYTE   0                                                       ;
                00  00FAB           .BYTE   0                                                       ;
              0000  00FAC           .WORD   0                                                       ;
              0000  00FAE           .WORD   0                                                       ;
              0000  00FB0           .WORD   0                                                       ;
              0000  00FB2           .WORD   0                                                       ;
              0000  00FB4           .WORD   0                                                       ;
              0000  00FB6           .WORD   0                                                       ;
              0000  00FB8           .WORD   0                                                       ;
              0000  00FBA           .WORD   0                                                       ;
              0000  00FBC           .WORD   0                                                       ;
              0000  00FBE           .WORD   0                                                       ;
              0000  00FC0           .WORD   0                                                       ;
                40  00FC2           .BYTE   64                                                      ;
                00  00FC3           .BYTE   0                                                       ;
                00  00FC4           .BYTE   0                                                       ;
                00  00FC5           .BYTE   0                                                       ;
                00  00FC6           .BYTE   0                                                       ;
                00  00FC7           .BYTE   0                                                       ;
                00  00FC8           .BYTE   0                                                       ;
                00  00FC9           .BYTE   0                                                       ;
              0000  00FCA           .WORD   0                                                       ;
        00000000    00FCC           .LONG   0                                                       ;
        00000000    00FD0           .LONG   0                                                       ;
                00  00FD4           .BYTE   0                                                       ;
                00  00FD5           .BYTE   0                                                       ;
                00  00FD6           .BYTE   0                                                       ;
                00  00FD7           .BYTE   0                                                       ;
                00  00FD8           .BYTE   0                                                       ;
                00  00FD9           .BYTE   0                                                       ;
                00  00FDA           .BYTE   0                                                       ;
                00  00FDB           .BYTE   0                                                       ;
                00  00FDC           .BYTE   0                                                       ;
                00  00FDD           .BYTE   0                                                       ;
                    00FDE           .BLKB   2                                                       ;
                13  00FE0 NAFPRO:   .BYTE   19                                                      ;
```

```
                58  00FE1          .BYTE   88
              0000  00FE2          .WORD   0
        00000000'   00FE4          .ADDRESS NAFKEY0
              FF00  00FE8          .WORD   -256
                00  00FEA          .BYTE   0
                00  00FEB          .BYTE   0
      0000      0000  00FEC        .WORD   C, 0
                00  00FF0          .BYTE   0
                00  00FF1          .BYTE   0
              0000  00FF2          .WORD   0
          00000000  00FF4          .LONG   0
          00000000  00FF8          .LONG   0
              0000  00FFC          .WORD   0
              0000  00FFE          .WORD   0
          00000000  01000          .LONG   0
          00000000  01004          .LONG   0
                    01008          .BLKB   48
                03  01038 NAFFAB: .BYTE   3
                50  01039          .BYTE   80
              0000  0103A          .WORD   0
          02000000  0103C          .LONG   33554432
          00000000  01040          .LONG   0
          00000000  01044          .LONG   0
          0000000A  01048          .LONG   10
              000A  0104C          .WORD   10
                0F  0104E          .BYTE   15
                0F  0104F          .BYTE   15
          00000000  01050          .LONG   0
                00  01054          .BYTE   0
                20  01055          .BYTE   32
                00  01056          .BYTE   0
                01  01057          .BYTE   1
          00000000  01058          .LONG   0
          00000000' 0105C          .ADDRESS NAFPRO
          00000000  01060          .LONG   0
          00000000' 01064          .ADDRESS P.ACE
          00000000' 01068          .ADDRESS P.ACF
                06  0106C          .BYTE   6
                04  0106D          .BYTE   4
              0064  0106E          .WORD   100
          00000000  01070          .LONG   0
              0000  01074          .WORD   0
                00  01076          .BYTE   0
                00  01077          .BYTE   0
          00000000  01078          .LONG   0
          00000000  0107C          .LONG   0
              0000  01080          .WORD   0
                00  01082          .BYTE   0
                00  01083          .BYTE   0
          00000000  01084          .LONG   0
                    01088 STATUS:  .BLKB   4

                                   .PSECT  $GLOBAL$,NOEXE,2

                    00000 DISBUF::.BLKB   132
          00000084  00084 DISDSC::.LONG   132
          00000000' 00088          .ADDRESS DISBUF
```

```
                        0008C FAODSC::.BLKB    8
                        00094 RABPTR::.BLKB    4
                        00098 UAF$GQ_SYSUAFF::
                                      .BLKB    8
                        000A0 UAF$GL_CTLMSK::
                                      .BLKB    8
                00000000  000A8 BY_ACCOUNT::
                                      .LONG    0                                    ;
                        000AC MATCH_TOKEN::
                                      .BLKB    66
                        000EE         .BLKB    2
                        000F0 MATCH_TOKENLEN::
                                      .BLKB    4
                        000F4 WILD_NETUSER::
                                      .BLKB    4
                00000000  000F8 CALL_COUNT::
                                      .LONG    0                                    ;
                      00# 000FC TOKENDSC::
                                      .BYTE    0[3]                                 ;
                      02  000FF         .BYTE    2                                    ;
                          00100         .BLKB    4
                          00104 CMDLINDSC::
                                      .BLKB    8
                          0010C NETUAF_EXISTS::
                                      .BLKB    4
                          00110 RDB_EXISTS::
                                      .BLKB    4
                          00114 RMSERR::.BLKB    4
                          00118 RIGHTSLIST_MODIFIED::
                                      .BLKB    1
                          00119         .BLKB    3
                          0011C RECBUF::.BLKB    1412
                          006A0 NETBUF::.BLKB    100
                    01    00704 UAFRAB::.BYTE    1                                    ;
                    44    00705         .BYTE    68                                   ;
                  0000    00706         .WORD    0                                    ;
                00000000  00708         .LONG    0                                    ;
                00000000  0070C         .LONG    0                                    ;
                00000000  00710         .LONG    0                                    ;
                  0000#   00714         .WORD    0[3]                                 ;
                  0000    0071A         .WORD    0                                    ;
                00000000  0071C         .LONG    0                                    ;
                  0000    00720         .WORD    0                                    ;
                    00    00722         .BYTE    0                                    ;
                    00    00723         .BYTE    0                                    ;
                  0584    00724         .WORD    1412                                 ;
                  0000    00726         .WORD    0                                    ;
                00000000  00728         .LONG    0                                    ;
                00000000  0072C         .LONG    0                                    ;
                00000000  00730         .LONG    0                                    ;
                00000000' 00734         .ADDRESS RECBUF+4                             ;
                    20    00738         .BYTE    32                                   ;
                    00    00739         .BYTE    0                                    ;
                    00    0073A         .BYTE    0                                    ;
                    00    0073B         .BYTE    0                                    ;
                00000000  0073C         .LONG    0                                    ;
                00000000' 00740         .ADDRESS UAFFAB                               ;
```

```
00000000  00744          .LONG   0
      0'  00748 NAFRAB::.BYTE   1
      44  00749          .BYTE   68
    0000  0074A          .WORD   0
00000000  0074C          .LONG   0
00000000  00750          .LONG   0
00000000  00754          .LONG   0
    0000# 00758          .WORD   0[3]
    0000  0075E          .WORD   0
00000000  00760          .LONG   0
    0000  00764          .WORD   0
      01  00766          .BYTE   1
      00  00767          .BYTE   0
    0064  00768          .WORD   100
    0064  0076A          .WORD   100
00000000  0076C          .LONG   0
00000000  00770          .LONG   0
00000000  00774          .LONG   0
00000000' 00778          .ADDRESS NETBUF
      40  0077C          .BYTE   64
      00  0077D          .BYTE   0
      00  0077E          .BYTE   0
      00  0077F          .BYTE   0
00000000  00780          .LONG   0
00000000' 00784          .ADDRESS NAFFAB
00000000  00788          .LONG   0
          0078C TIME_BUF::
                         .BLKB   8
          00794 PWD_FLAG::
                         .BLKB   4
      01  00798 OUTRAB::.BYTE   1
      44  00799          .BYTE   68
    0000  0079A          .WORD   0
00000000  0079C          .LONG   0
00000000  007A0          .LONG   0
00000000  007A4          .LONG   0
    0000# 007A8          .WORD   0[3]
    0000  007AE          .WORD   0
00000000  007B0          .LONG   0
    0000  007B4          .WORD   0
      00  007B6          .BYTE   0
      00  007B7          .BYTE   0
    0000  007B8          .WORD   0
    0000  007BA          .WORD   0
00000000  007BC          .LONG   0
00000000' 007C0          .ADDRESS DISBUF
00000000  007C4          .LONG   0
00000000  007C8          .LONG   0
      00  007CC          .BYTE   0
      00  007CD          .BYTE   0
      00  007CE          .BYTE   0
      00  007CF          .BYTE   0
00000000  007D0          .LONG   0
00000000' 007D4          .ADDRESS OUTFAB
00000000  007D8          .LONG   0
          007DC FOUND_MATCH::
                         .BLKB   4
```

```
007E0 UIC_FLAG::
                .BLKB   4
007E4 GRP_WILD::
                .BLKB   4
007E8 MEM_WILD::
                .BLKB   4
007EC STR_WILD::
                .BLKB   4

        SD_TOKEN1=              P.AAA
        SD_TOKEN2=              P.AAC
        SD_BRIEF=               P.AAE
        SD_FULL=                P.AAG
        SD_ADD_IDENTIFIER=      P.AAI
        SD_REMOVE_IDENTIFIER=
                                P.AAK
        SD_MODIFY_IDENTIFIER=
                                P.AAM
        ENCRYPT==               2
        DISBUFLEN==             132
        SYSUAF_STRING=          P.AAP
        NETUAF_STRING=          P.AAQ
        MOD_ACT_DSC=            P.AAR
        ADD_ACT_DSC=            P.AAT
        REM_ACT_DSC=            P.AAV
        FAO_LIN_DSC=            P.AAX
        DBL_COLON=              P.AAZ
        WAKEDELTA=              P.ABB
        DEFUSER=                P.ABC
        DEFPASS=                P.ABD
        DEFCLITABL=             P.ABE
        DEFACT=                 P.ABF
        DEFCLI=                 P.ABG
        DEFOWNER=               P.ABH
        DEFLGICMD=              P.ABI
        DEFGRP=                 128
        DEFMEM=                 128
        DEFDIR=                 P.ABJ
        DEFDEV=                 P.ABK
        DEFBIOLM=               6
        DEFBYTLM=               4096
        DEFDIOLM=               6
        DEFFILLM=               20
        DEFFLAGS=               0
        DEFTQCNT=               10
        DEFPRCCNT=              2
        DEFPRI=                 4
        DEFQUEPRI=              4
        DEFWSQUOTA=             200
        DEFDFWSCNT=             150
        DEFWSEXTENT=            500
        DEFCPUTIM=              0
        DEFASTLM=               10
        DEFPGFLQUOTA=           10000
        DEFENQLM=               10
        DEFPBYTLM=              0
        DEFSHRFILLM=            0
```

```
DEFMAXJOBS=         0
DEFMAXACCTJOBS=     0
DEFMAXDETACH=       0
DEFJTQUOTA=         1024
DEFPRIMEDAYS=       96
DEFHOURS=           0
DEFPRIV=            P.ABL
DEFPWDLENGTH=       6
DEFPWDLIFE=         P.ABM
SYSUSER=            P.ABN
SYSPASS=            P.ABO
SYSCLITABL=         P.ABP
SYSACT=             P.ABQ
SYSCLI=             P.ABR
SYSOWNER=           P.ABS
SYSLGICMD=          P.ABT
SYSGRP=             1
SYSMEM=             4
SYSDIR=             P.ABU
SYSDEV=             P.ABV
SYSBIOLM=           12
SYSBYTLM=           20480
SYSDIOLM=           12
SYSFILLM=           20
SYSFLAGS=           0
SYSTQCNT=           20
SYSPRCCNT=          10
SYSPRI=             4
SYSQUEPRI=          4
SYSWSQUOTA=         350
SYSWSEXTENT=        1024
SYSDFWSCNT=         150
SYSCPUTIM=          0
SYSASTLM=           20
SYSPGFLQUOTA=       10000
SYSENQLM=           20
SYSPBYTLM=          0
SYSSHRFILLM=        0
SYSMAXJOBS=         0
SYSMAXDETACH=       0
SYSJTQUOTA=         1024
SYSMAXACCTJOBS=     0
SYSPRIMEDAYS=       96
SYSHOURS=           0
SYSPRIV=            P.ABW
SYSPWDLENGTH=       8
SYSPWDLIFE=         P.ABX
TOKENLEN==          TOKENDSC
TOKENPTR==          TOKENDSC+4
REC_USER_DSC==      P.ACG
REC_ENCRYPT_DSC==   P.ACH
SYMBOL_STR==        P.ACI
ACCPRMPT=           P.ACJ
ACCPRMPT2=          P.ACK
NEWMSG20=           P.ACL
FOREIGN_CMDLINDSC=  P.ACM
        .EXTRN  CLI$_BUFOVF, CLI$_NOCLINT
```

```
        .EXTRN  LBR$OUTPUT_HELP
        .EXTRN  LIB$GET_FOREIGN
        .EXTRN  LIB$GET_INPUT, LIB$PUT_OUTPUT
        .EXTRN  FMG$MATCH_NAME, CLI$DCL_PARSE
        .EXTRN  CLI$DISPATCH, CLI$PRESENT
        .EXTRN  CLI$GET_VALUE, UPDATE_RECORD
        .EXTRN  PARSE_WILD, LGI$HPWD
        .EXTRN  UAF$ADD_IDENT_RECBUF
        .EXTRN  UAF$BUILD_HOLDER
        .EXTRN  UAF$FIND_OIC, UAF$REMOVE_IDENT_RECBUF
        .EXTRN  UAF$WRITE_RIGHTS
        .EXTRN  EXE$GL_SYSUIC, RDB_HEADER_FLAG
        .EXTRN  RDB_LIST_FLAG, ATTRIBUTES
        .EXTRN  HOLDER, IDENT, AUTHORIZE_COMMANDS
        .EXTRN  PRV$AB_NAMES, LIB$SIGNAL
        .EXTRN  UAF$_ADDERR, UAF$_ADDMSG
        .EXTRN  UAF$_BADNODFORM
        .EXTRN  UAF$_BADSPC, UAF$_BADUSR
        .EXTRN  UAF$_CLIWARNMSG
        .EXTRN  UAF$_CMDTOOLONG
        .EXTRN  UAF$_CONERR, UAF$_COPMSG
        .EXTRN  UAF$_CREERR, UAF$_DEFERR
        .EXTRN  UAF$_DEFPWD, UAF$_DONEMSG
        .EXTRN  UAF$_GETERR, UAF$_HELPERR
        .EXTRN  UAF$_INVCMD, UAF$_INVRSP
        .EXTRN  UAF$_INVUSERNAME
        .EXTRN  UAF$_KEYNOTFND, UAF$_KEYNOTUNQ
        .EXTRN  UAF$_LSTERR, UAF$_LSTMSG1
        .EXTRN  UAF$_LSTMSG2, UAF$_MDFYERR
        .EXTRN  UAF$_MDFYMSG, UAF$_NAFADDERR
        .EXTRN  UAF$_NAFADDMSG, UAF$_NAFAEX
        .EXTRN  UAF$_NAFCONERR, UAF$_NAFCREERR
        .EXTRN  UAF$_NAFDNE, UAF$_NAFDONEMSG
        .EXTRN  UAF$_NAFNOMODS, UAF$_NAFUAEERR
        .EXTRN  UAF$_NAMETOOBIG
        .EXTRN  UAF$_NETLSTMSG, UAF$_NAOFIL
        .EXTRN  UAF$_NAONAF, UAF$_NOARG
        .EXTRN  UAF$_NODEFPWD, UAF$_NODTOOBIG
        .EXTRN  UAF$_NOMODS, UAF$_NOTUNQ
        .EXTRN  UAF$_NOUSERNAME
        .EXTRN  UAF$_PREMMSG, UAF$_PUTERR
        .EXTRN  UAF$_RDBDONEMSG
        .EXTRN  UAF$_RDBMDFYERR
        .EXTRN  UAF$_RDBMDFYERRU
        .EXTRN  UAF$_RDBMDFYMSG
        .EXTRN  UAF$_RDBNOMODS, UAF$_REMDEF
        .EXTRN  UAF$_REMERR, UAF$_REMMSG
        .EXTRN  UAF$_REMSYS, UAF$_RENDEF
        .EXTRN  UAF$_RENMSG, UAF$_RENSYS
        .EXTRN  UAF$_RONLY, UAF$_SHOW_ERR
        .EXTRN  UAF$_SYSMSG1, UAF$_SYSMSG2
        .EXTRN  UAF$_UAEERR, UAF$_OICERR
        .EXTRN  UAF$_ZISQUAL, UAF$_ZZPRACREN
        .EXTRN  SYS$OPEN, SYS$CONNECT
        .EXTRN  SYS$CREATE

        .PSECT  $CODE$,NOWRT,2
```

```
                                  0FFC 00000 START:  .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 0902
                          5B 00000000G   00   9E 00002       MOVAB   AUTHORIZE_COMMANDS, R11
                          5A 00000000G   00   9E 00009       MOVAB   SYS$CONNECT, R10
                          59 00000000G   8F   D0 00010       MOVL    #CLI$_NOCLINT, R9
                          58 00000000G   00   9E 00017       MOVAB   LIB$STOP, R8
                          57 00000000'   00   9E 0001E       MOVAB   CMDLINDSC, R7
                          56 00000000'   00   9E 00025       MOVAB   STATUS, R6
                                14       A7   94 0002C       CLRB    RIGHTSLIST_MODIFIED                  ; 0947
                              EFC4       C6   7C 0002F       CLRQ    MODIFY_FLAG                          ; 0949
                              F9B8       C6   9F 00033       PUSHAB  INFAB                                ; 0955
                00000000G     00             FB 00037       CALLS   #1, SYS$OPEN
                              66             50 D0 0003E       MOVL    R0, STATUS
                          10  A7             50 D0 00041       MOVL    R0, RMSERR
                              05             50 E8 00045       BLBS    R0, 1$
                                             66 DD 00048       PUSHL   STATUS                               ; 0957
                              68             01 FB 0004A       CALLS   #1, LIB$STOP
                                        FA08 C6 9F 0004D 1$:   PUSHAB  INRAB                                ; 0959
                              6A             01 FB 00051       CALLS   #1, SYS$CONNECT
                              66             50 D0 00054       MOVL    R0, STATUS
                          10  A7             50 D0 00057       MOVL    R0, RMSERR
                              05             50 E8 0005B       BLBS    R0, 2$
                                             66 DD 0005E       PUSHL   STATUS                               ; 0961
                              68             01 FB 00060       CALLS   #1, LIB$STOP
                                        FA4C C6 9F 00063 2$:   PUSHAB  OUTFAB                               ; 0963
                00000000G     00             01 FB 00067       CALLS   #1, SYS$CREATE
                                        0694 C7 9F 0006E       PUSHAB  OUTRAB                               ; 0964
                              6A             01 FB 00072       CALLS   #1, SYS$CONNECT
                00000000V     00             00 FB 00075       CALLS   #0, SETUP                            ; 0966
                                             57 DD 0007C       PUSHL   R7                                   ; 0973
                                             7E D4 0007E       CLRL    -(SP)
                                  00000000'  00   9F 00080       PUSHAB  FOREIGN_CMDLINDSC
                          00000000G     00   03 FB 00086       CALLS   #3, LIB$GET_FOREIGN
                                        30   50 E9 0008D       BLBC    R0, 4$
                                             67 D5 00090       TSTL    CMDLINDSC
                                             2C 13 00092       BEQL    4$
                                  04    A7   F018 C6 9E 00094       MOVAB   CMDBUF, CMDLINDSC+4              ; 0979
                                       0880  8F BB 0009A       PUSHR   #^M<R7,R11>                          ; 0981
                          00000000G     00   02 FB 0009E       CALLS   #2, CLI$DCL_PARSE
                                        66   50 D0 000A5       MOVL    R0, STATUS
                                        09   50 E9 000A8       BLBC    R0, 3$
                          00000000G     00   00 FB 000AB       CALLS   #0, CLI$DISPATCH                     ; 0984
                                             43 11 000B2       BRB     7$                                   ; 0985
                                        59   66 D1 000B4 3$:   CMPL    STATUS, R9                           ; 0991
                                             37 13 000B7       BEQL    6$
                          00000000V     00   00 FB 000B9       CALLS   #0, ACC$EXIT                         ; 0995
                          00000000V     00   00 FB 000C0 4$:   CALLS   #0, GET_CMD_LINE                     ; 1008
                                        F6   50 E9 000C7       BLBC    R0, 4$
                                       0880  8F BB 000CA       PUSHR   #^M<R7,R11>                          ; 1010
                          00000000G     00   02 FB 000CE       CALLS   #2, CLI$DCL_PARSE
                                        66   50 D0 000D5       MOVL    R0, STATUS
                                        10   50 E9 000D8       BLBC    R0, 5$
08                  00                  6E   00 2C 000DB       MOVC5   #0, (SP), #0, #8, UAF$GL_CTLMSK      ; 1013
                                         9C  A7   000E0
                          00000000G     00   00 FB 000E2       CALLS   #0, CLI$DISPATCH                     ; 1014
                                             D5 11 000E9       BRB     4$                                   ; 1010
                                        59   66 D1 000EB 5$:   CMPL    STATUS, R9                           ; 1017
```

```
                                 50  12 000EE          BNEQ    4$                                                      ;
                                 59  DD 000F0 6$:      PUSHL   R9                                                      ; 1019
                              68 01  FB 000F2          CALLS   #1, LIB$STOP                                            ;
                                 C9  11 000F5          BRB     4$                                                      ; 1010
                              r0 01  D0 000F7 7$:      MOVL    #1, R0                                                  ; 1022
                                 04  000FA             RET                                                             ; 1024

; Routine Size:  251 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  933          1025  1 %sbttl 'setup - open initial files'
;  934          1026  1 routine setup : novalue =
;  935          1027  2 begin
;  936          1028  2
;  937          1029  2 !++
;  938          1030  2 !
;  939          1031  2 ! FUNCTIONAL DESCRIPTION:
;  940          1032  2 !
;  941          1033  2 !       This routine does all of the initial file manipulation for the
;  942          1034  2 !       program. It determines whether or not a previous SYSUAF.DAT exists.
;  943          1035  2 !       If not, it creates one (if the user wishes to proceed).
;  944          1036  2 !
;  945          1037  2 ! INPUTS:
;  946          1038  2 !
;  947          1039  2 !       none
;  948          1040  2 !
;  949          1041  2 ! IMPLICIT INPUTS:
;  950          1042  2 !
;  951          1043  2 !       none
;  952          1044  2 !
;  953          1045  2 ! OUTPUTS:
;  954          1046  2 !
;  955          1047  2 !       none
;  956          1048  2 !
;  957          1049  2 ! IMPLICIT OUTPUTS:
;  958          1050  2 !
;  959          1051  2 !       none
;  960          1052  2 !
;  961          1053  2 ! ROUTINE VALUE:
;  962          1054  2 !
;  963          1055  2 !       none
;  964          1056  2 !
;  965          1057  2 ! SIDE EFFECTS:
;  966          1058  2 !
;  967          1059  2 !       none
;  968          1060  2 !--
;  969          1061  2
;  970          1062  2 local
;  971          1063  2         status          : long,
;  972          1064  2         curpriv         : vector [2],
;  973          1065  2         procpriv        : vector [2],
;  974          1066  2         item_list       : block [64, byte],
;  975          1067  2         newfile;                        ! indicates new file must
;  976          1068  2                                         ! be created.
;  977          1069  2
;  978          1070  2 item_list [0,0,16,0] = 4;
;  979          1071  2 item_list [2,0,16,0] = jpi$_pid;
;  980          1072  2 item_list [4,0,32,0] = pid;
;  981          1073  2 item_list [8,0,32,0] = 0;
;  982          1074  2 item_list [12,0,16,0] = 12;
;  983          1075  2 item_list [14,0,16,0] = jpi$_username;
;  984          1076  2 item_list [16,0,32,0] = username_buf;
;  985          1077  2 item_list [20,0,32,0] = username_dsc[0];
;  986          1078  2 item_list [24,0,16,0] = 8;
;  987          1079  2 item_list [26,0,16,0] = jpi$_curpriv;
;  988          1080  2 item_list [28,0,32,0] = curpriv;
;  989          1081  2 item_list [32,0,32,0] = 0;
```

```
;  990     1082  2 item_list [36,0,16,0] = 8;
;  991     1083  2 item_list [38,0,16,0] = jpi$_procpriv;
;  992     1084  2 item_list [40,0,32,0] = procpriv;
;  993     1085  2 item_list [44,0,32,0] = 0;
;  994     1086  2 item_list [48,0,16,0] = 4;
;  995     1087  2 item_list [50,0,16,0] = jpi$_sts;
;  996     1088  2 item_list [52,0,32,0] = pcb_sts;
;  997     1089  2 item_list [56,0,32,0] = 0;
;  998     1090  2 item_list [60,0,32,0] = 0;
;  999     1091  2
; 1000     1092  2 $getjpi (itmlst = item_list);           ! Obtain pid, username and privs
; 1001     1093  2
; 1002     1094  2 username_dsc[0] = namelen (uaf$s_username, username_buf);
; 1003     1095  2
; 1004     1096  2 !**************************************************************************
; 1005     1097  2 !
; 1006     1098  2 !                              Open SYSUAF.DAT
; 1007     1099  2 !
; 1008     1100  2 !**************************************************************************
; 1009     1101  2
; 1010     1102  2 newfile = false;                        ! note no new file yet
; 1011     1103  2
; 1012     1104  3 if rmsbad ($open (fab = uaffab))
; 1013     1105  2 then
; 1014     1106  2 !
; 1015     1107  2 ! sysuaf.dat doesn't exist
; 1016     1108  2 !
; 1017     1109  3     begin
; 1018     1110  3     LIB$SIGNAL(UAF$_NAOFIL, 0, .rmserr);
; 1019     1111  3     if .rmserr eql rms$_fnf
; 1020     1112  3     then
; 1021     1113  4         begin
; 1022     1114  4         while true
; 1023     1115  4         do
; 1024     1116  4 !
; 1025     1117  4 ! Ask if a new one is desired
; 1026     1118  4 !
; 1027     1119  5             begin
; 1028     1120  5             ask (newmsg20, cmdbuf[0], cmdbuflen);
; 1029     1121  5             if .cmdbuf [0] eql 'Y'
; 1030     1122  5             then exitloop newfile = true;
; 1031     1123  5             if .cmdbuf [0] eql 'N'
; 1032     1124  5             then acc$exit ();
; 1033     1125  5             LIB$SIGNAL(UAF$_INVRSP);
; 1034     1126  4             end;
; 1035     1127  4         end
; 1036     1128  3     else
; 1037     1129  3         acc$exit ();
; 1038     1130  2     end;
; 1039     1131  2
; 1040     1132  2
; 1041     1133  2 !
; 1042     1134  2 ! The file will be created if it does not already exist.
; 1043     1135  2 ! In any case connect the RAB.
; 1044     1136  2 !
; 1045     1137  2
; 1046     1138  2 if .newfile
```

```
;  1047     1139  2 then
;  1048     1140  2 !
;  1049     1141  2 ! A new file is requested
;  1050     1142  2 !
;  1051     1143  3     if rmsbad ($create (fab = uaffab))
;  1052     1144  2     then
;  1053     1145  2 !
;  1054     1146  2 ! Quit regardless of error on a $CREATE: don't
;  1055     1147  2 ! want to give read-only user ability to create a file
;  1056     1148  2 !
;  1057     1149  2         LIB$SIGNAL(UAF$_CREERR, 0, .rmserr);
;  1058     1150  2
;  1059     1151  3 if rmsbad ($connect (rab = uafrab))
;  1060     1152  2 then LIB$SIGNAL(UAF$_CONERR, 0, .rmserr);
;  1061     1153  2 uafrab[rab$b_rac] = rab$c_key;          ! normal access is by key
;  1062     1154  2
;  1063     1155  2 !
;  1064     1156  2 ! Check to see if there was no old file to use.  If so write a default and
;  1065     1157  2 ! a system manager record.
;  1066     1158  2 !
;  1067     1159  2
;  1068     1160  2 if .newfile
;  1069     1161  2 then
;  1070     1162  3     begin
;  1071     1163  3     modify_flag = true;                ! must rename when done
;  1072     1164  3     build_ini_recs ();                 ! build default and system manager records
;  1073     1165  3     uafrab[rab$w_rsz] = uaf$c_fixed;
;  1074     1166  3
;  1075     1167  3     default_size = uaf$c_fixed;
;  1076     1168  3     uafrab[rab$l_rbf] = default_record; ! insert default record address
;  1077     1169  4     if rmsbad ($put (rab = uafrab))
;  1078     1170  3     then LIB$SIGNAL(UAF$_PUTERR, 0, .rmserr);   ! report error
;  1079     1171  3     uafrab[rab$l_rbf] = recbuf;        ! establish proper address (and
;  1080     1172  3                                        ! address of system record)
;  1081     1173  4     if rmsbad ($put (rab = uafrab))    ! output system record
;  1082     1174  3     then LIB$SIGNAL(UAF$_PUTERR, 0, .rmserr);
;  1083     1175  3     end
;  1084     1176  2 else
;  1085     1177  2 !
;  1086     1178  2 ! Read in the default record.
;  1087     1179  2 !
;  1088     1180  3     begin
;  1089     1181  3     uafrab[rab$l_ubf] = default_record;
;  1090     1182  3     if not locate_user (.defuser<0,8>, defuser+1, false)
;  1091     1183  3     then LIB$SIGNAL(UAF$_DEFERR, 0, .rmserr);
;  1092     1184  3     default_size = .uafrab[rab$w_rsz];
;  1093     1185  2     end;
;  1094     1186  2
;  1095     1187  2 uafrab[rab$l_ubf] = recbuf;             ! establish proper addresses
;  1096     1188  2 uafrab[rab$l_rbf] = recbuf;
;  1097     1189  2
;  1098     1190  2 !**************************************************************************
;  1099     1191  2 !
;  1100     1192  2 !                           Open NETUAF.DAT
;  1101     1193  2 !
;  1102     1194  2 !**************************************************************************
;  1103     1195  2
```

```
;  1104          1196  2 netuaf_exists = true;                    ! Assume NETUAF.DAT exists...
;  1105          1197  2
;  1106          1198  2 !
;  1107          1199  2 ! Try to open NETUAF.DAT and see what happens...
;  1108          1200  2 !
;  1109          1201  3 if rmsbad ($open (fab = naffab))
;  1110          1202  2 then
;  1111          1203  2 !
;  1112          1204  2 ! Couldn't open it
;  1113          1205  2 !
;  1114          1206  2     if .rmserr eql rms$_fnf
;  1115          1207  2     then
;  1116          1208  2         netuaf_exists = false          ! it doesn't exist
;  1117          1209  2     else
;  1118          1210  2         LIB$SIGNAL(UAF$_NAONAF, 0, .rmserr)     ! open error for some other reason
;  1119          1211  2
;  1120          1212  2 else
;  1121          1213  2 !
;  1122          1214  2 ! NETUAF.DAT opened without error
;  1123          1215  2 !
;  1124          1216  3     if rmsbad ($connect (rab = nafrab))
;  1125          1217  2     then LIB$SIGNAL(UAF$_NAFCONERR)             ! connect error
;  1126          1218  2 !
;  1127          1219  2 ! Everything opened and connected, establish proper NETUAF addresses
;  1128          1220  2 !
;  1129          1221  2     else
;  1130          1222  3         begin
;  1131          1223  3         nafrab [rab$l_ubf] = netbuf;
;  1132          1224  3         nafrab [rab$l_rbf] = netbuf;
;  1133          1225  2         end;
;  1134          1226  2
;  1135          1227  2 !
;  1136          1228  2 ! Check to see if the rights data base exists.  Try to translate an ID
;  1137          1229  2 ! and if we get a file not found error then it doesn't exist.
;  1138          1230  2 !
;  1139          1231  2 ident[uic$v_format] = uic$k_uic_format ;
;  1140          1232  2 ident[uic$v_group]  = 1 ;
;  1141          1233  2 ident[uic$v_member] = 4 ;
;  1142          1234  2 status = $idtoasc ( id = .ident ) ;
;  1143          1235  2 if .status eql rms$_fnf
;  1144          1236  2     then rdb_exists = false
;  1145          1237  2     else rdb_exists = true ;
;  1146          1238  2
;  1147          1239  1 end;


                                                    .EXTRN  SYS$GETJPI, SYS$PUT
                                                    .EXTRN  SYS$IDTOASC

                                  0FFC 00000 SETUP:     .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11         ; 1026
                5B 00000000G   00   9E 00002         MOVAB   SYS$PUT, R11                                ;
                5A 00000000G   00   9E 00009         MOVAB   SYS$CONNECT, R10                            ;
                59 00000000V   00   9E 00010         MOVAB   ACC$EXIT, R9                                ;
                58 00000000G   00   9E 00017         MOVAB   SYS$OPEN, R8                                ;
                57 00000000'   00   9E 0001E         MOVAB   NEWMSG20, R7                                ;
                56 00000000G   00   9E 00025         MOVAB   IDENT, R6                                   ;
```

```
                                 55 00000000G   00   9E 0002C          MOVAB   LIB$SIGNAL, R5                                  ;
                                 54 00000000'   00   9E 00033          MOVAB   USERNAME_BUF, R4                               ;
                                 53 00000000'   00   9E 0003A          MOVAB   RMSERR, R3                                     ;
                                 5E        B4   AE   9E 00041          MOVAB   -76(SP), SP                                    ;
                                    03190004   8F   DD 00045          PUSHL   #51970052                                      ; 1070
                              04    AE        24   A4   9E 0004B          MOVAB   PID, ITEM_LIST+4                               ; 1072
                                          08   AE   D4 00050          CLRL    ITEM_LIST+8                                    ; 1073
                              0C    AE 0202000C   8F   D0 00053          MOVL    #33685516, ITEM_LIST+12                        ; 1074
                              10    AE        64   9E 0005B          MOVAB   USERNAME_BUF, ITEM_LIST+16                     ; 1076
                              14    AE        28   A4   9E 0005F          MOVAB   USERNAME_DSC, ITEM_LIST+20                     ; 1077
                              18    AE 04000008   8F   D0 00064          MOVL    #67108872, ITEM_LIST+24                        ; 1078
                              1C    AE        48   AE   9E 0006C          MOVAB   CURPRIV, ITEM_LIST+28                          ; 1080
                                          20   AE   D4 00071          CLRL    ITEM_LIST+32                                   ; 1081
                              24    AE 02040008   8F   D0 00074          MOVL    #33816584, ITEM_LIST+36                        ; 1082
                              28    AE        40   AE   9E 0007C          MOVAB   PROCPRIV, ITEM_LIST+40                         ; 1084
                                          2C   AE   D4 00081          CLRL    ITEM_LIST+44                                   ; 1085
                              30    AE 03050004   8F   D0 00084          MOVL    #50659332, ITEM_LIST+48                        ; 1086
                              34    AE        20   A4   9E 0008C          MOVAB   PCB_STS, ITEM_LIST+52                          ; 1088
                                          38   AE   7C 00091          CLRQ    ITEM_LIST+56                                   ; 1089
                                               7E   7C 00094          CLRQ    -(SP)                                          ; 1092
                                               7E   D4 00096          CLRL    -(SP)                                          ;
                                          0C   AE   9F 00098          PUSHAB  ITEM_LIST                                      ;
                                               7E   7C 0009B          CLRQ    -(SP)                                          ;
                                               7E   D4 0009D          CLRL    -(SP)                                          ;
                          00000000G        00        07   FB 0009F          CALLS   #7, SYS$GETJPI                                 ;
                                64              20   3A 000A6          LOCC    #32, #32, USERNAME_BUF                         ; 1094
                           28   A4              20   50   C3 000AA          SUBL3   R0, #32, USERNAME_DSC                          ;
                                               52   D4 000AF          CLRL    NEWFILE                                        ; 1102
                                         0EF0   C4   9F 000B1          PUSHAB  UAFFAB                                         ; 1104
                                68              01   FB 000B5          CALLS   #1, SYS$OPEN                                   ;
                                63              50   D0 000B8          MOVL    R0, RMSERR                                     ;
                                4F              50   E8 000BB          BLBS    R0, 5$                                         ;
                                               63   DD 000BE          PUSHL   RMSERR                                         ; 1110
                                               7E   D4 000C0          CLRL    -(SP)                                          ;
                                    00000000G  8F   DD 000C2          PUSHL   #UAF$_NAOFIL                                   ;
                                65              03   FB 000C8          CALLS   #3, LIB$SIGNAL                                 ;
                          00018292   8F              63   D1 000CB          CMPL    RMSERR, #98962                                 ; 1111
                                               36   12 000D2          BNEQ    4$                                             ;
                                7E        0400   8F   3C 000D4 1$:      MOVZWL  #1024, -(SP)                                   ; 1120
                                          0098   C4   9F 000D9          PUSHAB  CMDBUF                                         ;
                                               57   DD 000DD          PUSHL   R7                                             ;
                          00000000V        00        03   FB 000DF          CALLS   #3, ASK                                        ;
                                50        0098   C4   9A 000E6          MOVZBL  CMDBUF, R0                                     ; 1121
                             59   8F              50   91 000EB          CMPB    R0, #89                                        ;
                                               05   12 000EF          BNEQ    2$                                             ;
                                52              01   D0 000F1          MOVL    #1, NEWFILE                                    ; 1122
                                               17   11 000F4          BRB     5$                                             ;
                             4E   8F              50   91 000F6 2$:      CMPB    R0, #78                                        ; 1123
                                               03   12 000FA          BNEQ    3$                                             ;
                                69              00   FB 000FC          CALLS   #0, ACC$EXIT                                   ; 1124
                                    00000000G  8F   DD 000FF 3$:      PUSHL   #UAF$_INVRSP                                   ; 1125
                                65              01   FB 00105          CALLS   #1, LIB$SIGNAL                                 ;
                                               CA   11 00108          BRB     1$                                             ; 1114
                                69              00   FB 0010A 4$:      CALLS   #0, ACC$EXIT                                   ; 1129
                                1E              52   E9 0010D 5$:      BLBC    NEWFILE, 6$                                    ; 1138
                                          0EF0   C4   9F 00110          PUSHAB  UAFFAB                                         ; 1143
                          00000000G        00        01   FB 00114          CALLS   #1, SYS$CREATE                                 ;
```

```
                        63              50  D0 0011B          MOVL    R0, RMSERR                                      ;
                        0D              50  E8 0011E          BLBS    R0, 6$                                          ;
                                        63  DD 00121          PUSHL   RMSERR                                          ; 1149
                                        7E  D4 00123          CLRL    -(SP)                                           ;
                    00000000G           8F  DD 00125          PUSHL   #UAF$_CREERR                                    ;
                        65              03  FB 0012B          CALLS   #3, LIB$SIGNAL                                  ;
                            05F0        C3  9F 0012E  6$:     PUSHAB  UAFRAB                                          ; 1151
                        6A              01  FB 00132          CALLS   #1, SYS$CONNECT                                 ;
                        63              50  D0 00135          MOVL    R0, RMSERR                                      ;
                        0D              50  E8 00138          BLBS    R0, 7$                                          ;
                                        63  DD 0013B          PUSHL   RMSERR                                          ; 1152
                                        7E  D4 0013D          CLRL    -(SP)                                           ;
                    00000000G           8F  DD 0013F          PUSHL   #UAF$_CONERR                                    ;
                        65              03  FB 00145          CALLS   #3, LIB$SIGNAL                                  ;
                060E    C3              01  90 00148  7$:     MOVB    #1, UAFRAB+30                                   ; 1153
                        5C              52  E9 0014D          BLBC    NEWFILE, 9$                                     ; 1160
                  44    A4              01  D0 00150          MOVL    #1, MODIFY_FLAG                                 ; 1163
           00000000V    00              00  FB 00154          CALLS   #0, BUILD_INI_RECS                              ; 1164
                0612    C3      0284    8F  B0 0015B          MOVW    #644, UAFRAB+34                                 ; 1165
                0498    C4      0284    8F  B0 00162          MOVW    #644, DEFAULT_SIZE                              ; 1167
                0618    C3      049C    C4  9E 00169          MOVAB   DEFAULT_RECORD, UAFRAB+40                       ; 1168
                            05F0        C3  9F 00170          PUSHAB  UAFRAB                                          ; 1169
                        6B              01  FB 00174          CALLS   #1, SYS$PUT                                     ;
                        63              50  D0 00177          MOVL    R0, RMSERR                                      ;
                        0D              50  E8 0017A          BLBS    R0, 8$                                          ;
                                        63  DD 0017D          PUSHL   RMSERR                                          ; 1170
                                        7E  D4 0017F          CLRL    -(SP)                                           ;
                    00000000G           8F  DD 00181          PUSHL   #UAF$_PUTERR                                    ;
                        65              03  FB 00187          CALLS   #3, LIB$SIGNAL                                  ;
                0618    C3        08    A3  9E 0018A  8$:     MOVAB   RECBUF, UAFRAB+40                               ; 1171
                            05F0        C3  9F 00190          PUSHAB  UAFRAB                                          ; 1173
                        6B              01  FB 00194          CALLS   #1, SYS$PUT                                     ;
                        63              50  D0 00197          MOVL    R0, RMSERR                                      ;
                        3F              50  E8 0019A          BLBS    R0, 11$                                         ;
                                        63  DD 0019D          PUSHL   RMSERR                                          ; 1174
                                        7E  D4 0019F          CLRL    -(SP)                                           ;
                    00000000G           8F  DD 001A1          PUSHL   #UAF$_PUTERR                                    ;
                        65              03  FB 001A7          CALLS   #3, LIB$SIGNAL                                  ;
                                        30  11 001AA          BRB     11$                                             ; 1160
                0614    C3      049C    C4  9E 001AC  9$:     MOVAB   DEFAULT_RECORD, UAFRAB+36                       ; 1181
                                        7E  D4 001B3          CLRL    -(SP)                                           ; 1182
                            FEFB        C7  9F 001B5          PUSHAB  DEFUSER+1                                       ;
                        7E  FEFA        C7  9A 001B9          MOVZBL  DEFUSER, -(SP)                                  ;
           00000000V    00              03  FB 001BE          CALLS   #3, LOCATE_USER                                 ;
                        0D              50  E8 001C5          BLBS    R0, 10$                                         ;
                                        63  DD 001C8          PUSHL   RMSERR                                          ; 1183
                                        7E  D4 001CA          CLRL    -(SP)                                           ;
                    00000000G           8F  DD 001CC          PUSHL   #UAF$_DEFERR                                    ;
                        65              03  FB 001D2          CALLS   #3, LIB$SIGNAL                                  ;
                0498    C4      0612    C3  B0 001D5  10$:    MOVW    UAFRAB+34, DEFAULT_SIZE                         ; 1184
                0614    C3        08    A3  9E 001DC  11$:    MOVAB   RECBUF, UAFRAB+36                               ; 1187
                0618    C3        08    A3  9E 001E2          MOVAB   RECBUF, UAFRAB+40                               ; 1188
                  F8    A3              01  D0 001E8          MOVL    #1, NETUAF_EXISTS                               ; 1196
                            1030        C4  9F 001EC          PUSHAB  NAFFAB                                          ; 1201
                        68              01  FB 001F0          CALLS   #1, SYS$OPEN                                    ;
                        63              50  D0 001F3          MOVL    R0, RMSERR                                      ;
                        20              50  E8 001F6          BLBS    R0, 13$                                         ;
```

```
                                50                      63   D0 001F9          MOVL    RMSERR, R0                                  ; 1206
                      00018292  8F                      50   D1 001FC          CMPL    R0, #98962
                                                         05   12 00203          BNEQ    12$
                                          F8            A3   D4 00205          CLRL    NETUAF_EXISTS                               ; 1208
                                                         35   11 00208          BRB     15$
                                                         50   DD 0020A 12$:     PUSHL   R0                                          ; 1210
                                                         7E   D4 0020C          CLRL    -(SP)
                                00000000G               8F   DD 0020E          PUSHL   #UAF$_NAONAF
                                65                      03   FB 00214          CALLS   #3, LIB$SIGNAL
                                                         26   11 00217          BRB     15$                                         ; 1206
                                          0634          C3   9F 00219 13$:     PUSHAB  NAFRAB                                      ; 1216
                                6A                      01   FB 0021D          CALLS   #1, SYS$CONNECT
                                63                      50   D0 00220          MOVL    R0, RMSERR
                                0B                      50   E8 00223          BLBS    R0, 14$
                                00000000G               8F   DD 00226          PUSHL   #UAF$_NAFCONERR                             ; 1217
                                65                      01   FB 0022C          CALLS   #1, LIB$SIGNAL
                                                         0E   11 0022F          BRB     15$
                          0658  C3        058C          C3   9E 00231 14$:     MOVAB   NETBUF, NAFRAB+36                           ; 1223
                          065C  C3        058C          C3   9E 00238          MOVAB   NETBUF, NAFRAB+40                           ; 1224
                            03  A6          C0          8F   8A 0023F 15$:     BICB2   #192, IDENT+3                               ; 1231
    02    A6              0E                00          01   F0 00244          INSV    #1, #0, #14, IDENT+2                        ; 1232
                                66                      04   B0 0024A          MOVW    #4, IDENT                                   ; 1233
                                                         7E   7C 0024D          CLRQ    -(SP)                                       ; 1234
                                                         7E   7C 0024F          CLRQ    -(SP)
                                                         7E   D4 00251          CLRL    -(SP)
                                                         66   DD 00253          PUSHL   IDENT
                      00000000G  00                      06   FB 00255          CALLS   #6, SYS$IDTOASC
                      00018292  8F                      50   D1 0025C          CMPL    STATUS, #98962                              ; 1235
                                                         04   12 00263          BNEQ    16$
                                          FC            A3   D4 00265          CLRL    RDB_EXISTS                                  ; 1236
                                                         04   00268          RET
                            FC  A3                       01   D0 00269 16$:     MOVL    #1, RDB_EXISTS                              ; 1237
                                                         04   0026D          RET                                                 ; 1239

; Routine Size:  622 bytes,    Routine Base:  $CODE$ + 00FB
```

```
;1149     1240  1 %sbttl 'add_uaf - insert new user record'
;1150     1241  1 global routine add_uaf : novalue =
;1151     1242  2 begin
;1152     1243  2
;1153     1244  2 !++
;1154     1245  2 !
;1155     1246  2 ! FUNCTIONAL DESCRIPTION:
;1156     1247  2 !
;1157     1248  2 !       Routine to add new user to authorization file.
;1158     1249  2 !
;1159     1250  2 ! INPUTS:
;1160     1251  2 !
;1161     1252  2 !       none
;1162     1253  2 !
;1163     1254  2 ! IMPLICIT INPUTS:
;1164     1255  2 !
;1165     1256  2 !       none
;1166     1257  2 !
;1167     1258  2 ! OUTPUTS:
;1168     1259  2 !
;1169     1260  2 !       none
;1170     1261  2 !
;1171     1262  2 ! IMPLICIT OUTPUTS:
;1172     1263  2 !
;1173     1264  2 !       none
;1174     1265  2 !
;1175     1266  2 ! ROUTINE VALUE:
;1176     1267  2 !
;1177     1268  2 !       none
;1178     1269  2 !
;1179     1270  2 ! SIDE EFFECTS:
;1180     1271  2 !
;1181     1272  2 !       A record is added to SYSUAF.DAT
;1182     1273  2 !--
;1183     1274  2
;1184     1275  2 map
;1185     1276  2         tokenptr        : ref vector [,byte];
;1186     1277  2
;1187     1278  2 local
;1188     1279  2         user_dsc        : vector [2];   ! descriptor for username in record
;1189     1280  2
;1190     1281  2 !
;1191     1282  2 ! Make sure a username was specified.
;1192     1283  2 !
;1193     1284  2
;1194     1285  2 if not cli$present (sd_token1)
;1195     1286  2 or not cli$get_value (sd_token1,tokendsc)
;1196     1287  2 or .tokenlen eql 0
;1197     1288  2 then return LIB$SIGNAL(UAF$_NOUSERNAME);
;1198     1289  2
;1199     1290  2 !
;1200     1291  2 ! ADD must check that the username supplied is not too long.
;1201     1292  2 !
;1202     1293  2 !***if .tokenlen gtr  uaf$s_username
;1203     1294  2 if .tokenlen gtr 12
;1204     1295  2 then return LIB$SIGNAL(UAF$_NAMETOOBIG);
;1205     1296  2
```

```
;  1206          1297  2 !
;  1207          1298  2 ! Make sure a legal username was entered, otherwise the account may not be
;  1208          1299  2 ! accessible via LOGIN or the Input Symbiont.
;  1209          1300  2 !
;  1210          1301  2 incru i to .tokenlen - 1
;  1211          1302  2 do
;  1212          1303  2     if ch$fail (ch$find_ch (.symbol_str<0,8>,
;  1213          1304  2                             symbol_str + 1,
;  1214          1305  2                             .tokenptr [.i]))
;  1215          1306  2     then return LIB$SIGNAL(UAF$_INVUSERNAME);
;  1216          1307  2 user_dsc[0] = .tokenlen;
;  1217          1308  2 user_dsc[1] = recbuf[uaf$t_username];
;  1218          1309  2
;  1219          1310  2 !
;  1220          1311  2 ! Move the default record to the current record buffer, so that
;  1221          1312  2 ! fields which are not entered will receive the default
;  1222          1313  2 ! value.  Then insert the username just entered.
;  1223          1314  2 !
;  1224          1315  2
;  1225          1316  2 ch$move (.default_size, default_record, recbuf);
;  1226          1317  2 ch$copy (.tokenlen, .tokenptr, ' ', uaf$s_username, recbuf[uaf$t_username]);
;  1227          1318  2
;  1228          1319  2 !
;  1229          1320  2 ! Call routine to fill in all supplied values.  Exit if any errors
;  1230          1321  2 ! were found.
;  1231          1322  2 !
;  1232          1323  2
;  1233          1324  2 pwd_flag = true;                          ! plan to supply a password
;  1234          1325  2 uafrab[rab$w_rsz] = .default_size;
;  1235          1326  2 uaf$gl_ctlmsk[uaf$v_add] = true;
;  1236          1327  2 if not update_record ()
;  1237          1328  2 then
;  1238          1329  3     begin
;  1239          1330  3     uaf$gl_ctlmsk[uaf$v_add] = false;
;  1240          1331  3     return;
;  1241          1332  2     end;
;  1242          1333  2
;  1243          1334  2 uaf$gl_ctlmsk[uaf$v_add] = false;
;  1244          1335  2
;  1245          1336  2 if .pwd_flag                              ! if no explicit password
;  1246          1337  2 then
;  1247          1338  3     begin
;  1248          1339  3     pwddsc[dsc$w_length] = .defpass<0,8>;
;  1249          1340  3     pwddsc[dsc$a_pointer] = defpass+1;
;  1250          1341  3     $gettim (timadr = time_buf);
;  1251          1342  3     recbuf[uaf$w_salt] = .time_buf<3*8,16>;
;  1252          1343  3     recbuf[uaf$b_encrypt] = encrypt;
;  1253          1344  3     lgi$hpwd (rec_encrypt_dsc, pwddsc, .recbuf[uaf$b_encrypt],
;  1254          1345  3                 .recbuf[uaf$w_salt], user_dsc);
;  1255          1346  2     end;
;  1256          1347  2
;  1257          1348  2 !
;  1258          1349  2 ! Now output the new record.
;  1259          1350  2 !
;  1260          1351  2
;  1261          1352  3 if rmsbad ($put (rab = uafrab))
;  1262          1353  2 then
```

```
;  1263          1354  2     if .rmserr eql rms$_dup
;  1264          1355  2     then return LIB$SIGNAL(UAF$_UAEERR)
;  1265          1356  2     else LIB$SIGNAL(UAF$_ADDERR, 0, .rmserr)
;  1266          1357  2 else
;  1267          1358  3     begin
;  1268          1359  3
;  1269          1360  3 !
;  1270          1361  3 ! Tell user that addition was successful.  Note that file was changed.
;  1271          1362  3 !
;  1272          1363  3     LIB$SIGNAL(UAF$_ADDMSG);
;  1273          1364  4     if (.uaf$gl_ctlmsk[uaf$v_cli]
;  1274          1365  4     and (not .uaf$gl_ctlmsk[uaf$v_clitables]))
;  1275          1366  3     then LIB$SIGNAL(UAF$_CLIWARNMSG);
;  1276          1367  3     security_audit (nsa$k_recid_sysuaf_add);
;  1277          1368  3     modify_flag = true;
;  1278          1369  4     if (cli$present ( sd_add_identifier ) and
;  1279          1370  4         .rdb_exists )
;  1280          1371  3     then
;  1281          1372  4         begin
;  1282          1373  4         !
;  1283          1374  4         ! Add the appropriate identifiers.
;  1284          1375  4         ! Set the resource attribute if /ATTRIB=RESOURCE was specified
;  1285          1376  4         ! and then add the appropriate identifiers to the rights data base
;  1286          1377  4         !
;  1287          1378  4         if cli$present (sd_attribresource)
;  1288          1379  4             then attributes = kgb$m_resource
;  1289          1380  4             else attributes = 0 ;
;  1290          1381  4         uaf$add_ident_recbuf () ;
;  1291          1382  3         end ;
;  1292          1383  2     end;
;  1293          1384  1 end;

                                                    .EXTRN  SYS$GETTIM

                                        0FFC 00000  .ENTRY  ADD_UAF, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-  ; 1241
                                                            R11
                    5B 00000000G   00   9E 00002    MOVAB   LIB$SIGNAL, R11
                    5A 00000000'   00   9E 00009    MOVAB   PWDDSC, R10
                    59 00000000'   00   9E 00010    MOVAB   SD_TOKEN1, R9
                    58 00000000'   00   9E 00017    MOVAB   TOKENDSC, R8
                    5E             08   C2 0001E    SUBL2   #8, SP
                                   59   DD 00021    PUSHL   R9                                          ; 1285
          00000000G 00             01   FB 00023    CALLS   #1, CLI$PRESENT
                    12             50   E9 0002A    BLBC    R0, 1$
                                   58   DD 0002D    PUSHL   R8                                          ; 1286
                                   59   DD 0002F    PUSHL   R9
          00000000G 00             02   FB 00031    CALLS   #2, CLI$GET_VALUE
                    04             50   E9 00038    BLBC    R0, 1$
                                   68   B5 0003B    TSTW    TOKENLEN                                    ; 1287
                                   08   12 0003D    BNEQ    2$
                       0C 00000000G 8F   DD 0003F 1$: PUSHL   #UAF$_NOUSERNAME                         ; 1288
                                   34   11 00045    BRB     6$
                    0C             68   B1 00047 2$: CMPW    TOKENLEN, #12                               ; 1294
                                   08   1B 0004A    BLEQU   3$
                       00000000G   8F   DD 0004C    PUSHL   #UAF$_NAMETOOBIG                            ; 1295
```

```
                                                  27  11 00052          BRB     6$                                                      ;
                                            53    68  3C 00054 3$:      MOVZWL  TOKENLEN, R3                                            ; 1301
                                                  53  D7 00057          DECL    R3                                                      ;
                                                  52  D4 00059          CLRL    I                                                       ;
                                                  23  11 0005B          BRB     8$                                                      ;
                                            51 01EC C9  9A 0005D 4$:      MOVZBL  SYMBOL_STR, R1                                          ; 1303
                                            50   04 A8  D0 00062          MOVL    TOKENPTR, R0                                            ; 1305
                                      01ED  C9      51    6240  3A 00066          LOCC    (I)[R0], R1, SYMBOL_STR+1                       ;
                                                  02  12 0006D          BNEQ    5$                                                      ;
                                                  51  D4 0006F          CLRL    R1                                                      ;
                                                  51  D5 00071 5$:      TSTL    R1                                                      ;
                                                  09  12 00073          BNEQ    7$                                                      ;
                                          00000000G   8F  DD 00075          PUSHL   #UAF$_INVUSERNAME                                       ; 1306
                                                00AF  31  00078 6$:      BRW     11$                                                     ;
                                                  52  D6 0007E 7$:      INCL    I                                                       ; 1305
                                            53    52  D1 00080 8$:      CMPL    I, R3                                                   ;
                                                  D8  1B 00083          BLEQU   4$                                                      ;
                                            57    68  3C 00085          MOVZWL  TOKENLEN, R7                                            ; 1307
                                            6E    57  D0 00088          MOVL    R7, USER_DSC                                            ;
                                      04    AE   24   A8  9E 0008B          MOVAB   RECBUF+4, USER_DSC+4                                    ; 1308
                                            56 FA78   CA  3C 00090          MOVZWL  DEFAULT_SIZE, R6                                        ; 1316
                                   20 A8    FA7C      CA  28 00095          MOVC3   R6, DEFAULT_RECORD, RECBUF                              ;
                                            50   04 A8  D0 0009C          MOVL    TOKENPTR, R0                                            ; 1317
                                20        20    60        57  2C 000A0          MOVC5   R7, (R0), #32, #32, RECBUF+4                    ;
                                                  24 A8       000A5                                                                   ;
                                         0698 C8        01  D0 000A7          MOVL    #1, PWD_FLAG                                    ; 1324
                                         062A C8        56  B0 000AC          MOVW    R6, UAFRAB+34                                   ; 1325
                                           A4 A8        02  88 000B1          BISB2   #2, UAF$GL_CTLMSK                               ; 1326
                                   00000000G  00        00  FB 000B5          CALLS   #0, UPDATE_RECORD                               ; 1327
                                            05        50  E8 000BC          BLBS    R0, 9$                                          ;
                                           A4 A8        02  8A 000BF          BICB2   #2, UAF$GL_CTLMSK                               ; 1330
                                                        04  000C3          RET                                                     ; 1329
                                           A4 A8        02  8A 000C4 9$:      BICB2   #2, UAF$GL_CTLMSK                               ; 1334
                                            3B      0698 C8  E9 000C8          BLBC    PWD_FLAG, TO$                                   ; 1336
                                            6A      0120 C9  9B 000CD          MOVZBW  DEFPASS, PWDDSC                                 ; 1339
                                           04 AA    0121 C9  9E 000D2          MOVAB   DEFPASS+1, PWDDSC+4                             ; 1340
                                                    0690 C8  9F 000D8          PUSHAB  TIME_BUF                                        ; 1341
                                   00000000G  00        01  FB 000DC          CALLS   #1, SYS$GETTIM                                  ;
                                         0186 C8    0693 C8  B0 000E3          MOVW    TIME_BUF+3, RECBUF+358                          ; 1342
                                         0188 C8         02  90 000EA          MOVB    #2, RECBUF+360                                  ; 1343
                                                         5E  DD 000EF          PUSHL   SP                                              ; 1344
                                            7E      0186 C8  3C 000F1          MOVZWL  RECBUF+358, -(SP)                               ; 1345
                                            7E      0188 C8  9A 000F6          MOVZBL  RECBUF+360, -(SP)                               ; 1344
                                                         5A  DD 000FB          PUSHL   R10                                             ;
                                                    01E4 C9  9F 000FD          PUSHAB  REC_ENCRYPT_DSC                                 ;
                                   00000000G  00        05  FB 00101          CALLS   #5, LGI$HPWD                                    ;
                                                    0608 C8  9F 00108 10$:     PUSHAB  UAFRAB                                          ; 1352
                                   00000000G  00        01  FB 0010C          CALLS   #1, SYS$PUT                                     ;
                                         18 A8          50  D0 00113          MOVL    R0, RMSERR                                      ;
                                            25          50  E8 00117          BLBS    R0, 13$                                         ;
                                            50      18 A8  D0 0011A          MOVL    RMSERR, R0                                      ; 1354
                                   000184EC 8F          50  D1 0011E          CMPL    R0, #99564                                      ;
                                                        0A  12 00125          BNEQ    12$                                             ;
                                                00000000G  8F  DD 00127          PUSHL   #UAF$_UAEERR                                    ; 1355
                                            6B          01  FB 0012D 11$:     CALLS   #1, LIB$SIGNAL                                  ;
                                                        04  00130          RET                                                     ;
                                                        50  DD 00131 12$:     PUSHL   R0                                              ; 1356
```

```
                                                7E  D4 00133          CLRL    -(SP)                                                       ;
                                    00000000G   8F  DD 00135          PUSHL   #UAF$_ADDERR                                                ;
                                          6B        03  FB 0013B          CALLS   #3, LIB$SIGNAL                                              ;
                                                    04  0013E          RET                                                                 ; 1354
                                    00000000G   8F  DD 0013F 13$:      PUSHL   #UAF$_ADDMSG                                                ; 1363
                                          6B        01  FB 00145          CALLS   #1, LIB$SIGNAL                                              ;
                              0E        A4    A8        03  E1 00148          BBC     #3, UAF$GL_CTLMSK, 14$                              ; 1364
                              09        A4    A8        04  E0 0014D          BBS     #4, UAF$GL_CTLMSK, 14$                              ; 1365
                                    00000000G   8F  DD 00152          PUSHL   #UAF$_CLIWARNMSG                                            ; 1366
                                          6B        01  FB 00158          CALLS   #1, LIB$SIGNAL                                              ;
                                    00010002    8F  DD 0015B 14$:      PUSHL   #65538                                                      ; 1367
                                 00000000V      00        01  FB 00161          CALLS   #1, SECURITY_AUDIT                                          ;
                                     F624       CA        01  D0 00168          MOVL    #1, MODIFY_FLAG                                     ; 1368
                                          44    A9  9F 0016D          PUSHAB  SD_ADD_IDENTIFIER                                           ; 1369
                                 00000000G      00        01  FB 00170          CALLS   #1, CLI$PRESENT                                             ;
                                          28        50  E9 00177          BLBC    R0, 17$                                                     ;
                                          24    14  A8  E9 0017A          BLBC    RDB_EXISTS, 17$                                     ; 1370
                                        F5D8    CA  9F 0017E          PUSHAB  SD_ATTRIBRESOURCE                                           ; 1378
                                 00000000G      00        01  FB 00182          CALLS   #1, CLI$PRESENT                                             ;
                                          09        50  E9 00189          BLBC    R0, 15$                                                     ;
                                 00000000G      00        01  D0 0018C          MOVL    #1, ATTRIBUTES                                      ; 1379
                                                    06  11 00193          BRB     16$                                                         ;
                                    00000000G   00  D4 00195 15$:      CLRL    ATTRIBUTES                                                  ; 1380
                                 00000000G      00        00  FB 0019B 16$:      CALLS   #0, UAF$ADD_IDENT_RECBUF                            ; 1381
                                                    04  001A2 17$:      RET                                                                 ; 1384

; Routine Size:  419 bytes,    Routine Base:  $CODE$ + 0369
```

```
;  1295     1385  1 %sbttl 'add_proxy - insert new proxy record'
;  1296     1386  1 global routine add_proxy : novalue =
;  1297     1387  2 begin
;  1298     1388  2
;  1299     1389  2 !++
;  1300     1390  2 !
;  1301     1391  2 ! FUNCTIONAL DES'RIPTION:
;  1302     1392  2 !
;  1303     1393  2 !       This routine adds an entry to the NETUAF.DAT Proxy Login File
;  1304     1394  2 !
;  1305     1395  2 ! INPUTS:
;  1306     1396  2 !
;  1307     1397  2 !       none
;  1308     1398  2 !
;  1309     1399  2 ! OUTPUTS:
;  1310     1400  2 !
;  1311     1401  2 !       none
;  1312     1402  2 !
;  1313     1403  2 ! IMPLICIT INPUTS:
;  1314     1404  2 !
;  1315     1405  2 !       TOKENLEN, TOKENPTR
;  1316     1406  2 !
;  1317     1407  2 ! IMPLICIT OUTPUTS:
;  1318     1408  2 !
;  1319     1409  2 !       none
;  1320     1410  2 !
;  1321     1411  2 ! ROUTINE VALUE:
;  1322     1412  2 !
;  1323     1413  2 !       none
;  1324     1414  2 !
;  1325     1415  2 ! SIDE EFFECTS:
;  1326     1416  2 !
;  1327     1417  2 !       A record is added to NETUAF.DAT
;  1328     1418  2 !
;  1329     1419  2 !--
;  1330     1420  2
;  1331     1421  2 local
;  1332     1422  2         node_len,
;  1333     1423  2         node_ptr,
;  1334     1424  2         remuser_len,
;  1335     1425  2         remuser_ptr,
;  1336     1426  2         locuser_len,
;  1337     1427  2         locuser_ptr;
;  1338     1428  2
;  1339     1429  2 !
;  1340     1430  2 ! Can't do anything if there is no NETUAF.DAT...
;  1341     1431  2 !
;  1342     1432  2 if not .netuaf_exists
;  1343     1433  2 then return LIB$SIGNAL(UAF$_NAFDNE);
;  1344     1434  2
;  1345     1435  2 !
;  1346     1436  2 ! Clear NETUAF.DAT buffer
;  1347     1437  2 !
;  1348     1438  2 ch$fill (' ', naf$c_length, netbuf);
;  1349     1439  2
;  1350     1440  2 !
;  1351     1441  2 ! Retrieve token from command line
```

```
;1352    1442  2 !
;1353    1443  2 cli$get_value (sd_token1, tokendsc);
;1354    1444  2
;1355    1445  2 !
;1356    1446  2 ! Make sure entry is in proper node::remoteuser format
;1357    1447  2 !
;1358    1448  2 if not remote_parse (node_ptr, node_len, remuser_ptr, remuser_len)
;1359    1449  2 then return;
;1360    1450  2 !
;1361    1451  2 ! Fill in NETBUF with new remotename field
;1362    1452  2 !
;1363    1453  2 ch$copy (.node_len, .node_ptr, ' ', naf$s_node, netbuf[naf$t_node]);
;1364    1454  2 ch$copy (.remuser_len, .remuser_ptr, ' ', naf$s_remuser, netbuf[naf$t_remuser]);
;1365    1455  2
;1366    1456  2 !
;1367    1457  2 ! Now get second token, the local user name
;1368    1458  2 !
;1369    1459  2 cli$get_value (sd_token2, tokendsc);
;1370    1460  2
;1371    1461  2 locuser_len = .tokenlen;
;1372    1462  2 locuser_ptr = .tokenptr;
;1373    1463  2 !***if .locuser_len gtru naf$s_localuser
;1374    1464  2 if .locuser_len gtru 12
;1375    1465  2 then return LIB$SIGNAL(UAF$_NAMETOOBIG);
;1376    1466  2
;1377    1467  2 !
;1378    1468  2 ! If local name is *, then use same name as remote user
;1379    1469  2 !
;1380    1470  2 if .tokenlen eql 1  and  . (.tokenptr)<0,8> eql '*'
;1381    1471  2 then
;1382    1472  3     begin
;1383    1473  3     locuser_len = .remuser_len;
;1384    1474  3     ch$move (naf$s_remuser, netbuf[naf$t_remuser], netbuf[naf$t_localuser]);
;1385    1475  3     end
;1386    1476  3 !
;1387    1477  3 ! Otherwise just copy into localuser field in NETBUF
;1388    1478  3 !
;1389    1479  2 else
;1390    1480  2     ch$copy (.locuser_len, .locuser_ptr, ' ',
;1391    1481  2              naf$s_localuser, netbuf[naf$t_localuser]);
;1392    1482  2
;1393    1483  2 !
;1394    1484  2 ! Make sure that the local user does indeed exist in SYSUAF.DAT
;1395    1485  2 ! (unless local user is *)
;1396    1486  2 !
;1397    1487  2 if not locate_user (.locuser_len, netbuf[naf$t_localuser], 0)
;1398    1488  3 and not (.locuser_len eql 1 and . (netbuf[naf$t_localuser])<0,8> eql '*')
;1399    1489  2 then return LIB$SIGNAL(UAF$_BADUSR, 2, .locuser_len, netbuf[naf$t_localuser]);
;1400    1490  2
;1401    1491  2 nafrab[rab$w_rsz] = naf$c_length;
;1402    1492  2
;1403    1493  2 !
;1404    1494  2 ! Add NETUAF.DAT record
;1405    1495  2 !
;1406    1496  3 if rmsbad ($put (rab = nafrab))
;1407    1497  2 then
;1408    1498  3     begin
```

```
;  1409          1499  3     if .rmserr eql rms$_dup
;  1410          1500  3     then
;  1411          1501  3         return LIB$SIGNAL(UAF$_NAFUAEERR)
;  1412          1502  3     else
;  1413          1503  3         LIB$SIGNAL(UAF$_NAFADDERR, 0, .rmserr)
;  1414          1504  3     end
;  1415          1505  2 else
;  1416          1506  2     begin
;  1417          1507  2     LIB$SIGNAL(UAF$_NAFADDMSG);
;  1418          1508  2     security_audit (nsa$k_recid_netuaf_add);
;  1419          1509  2     end;
;  1420          1510  2
;  1421          1511  2 netuaf_modified = true;
;  1422          1512  1 end;


                                                07FC 00000          .ENTRY  ADD_PROXY, Save R2,R3,R4,R5,R6,R7,R8,R9,R10 ; 1386
                                    5A 00000000G  00  9E 00002      MOVAB   CLI$GET_VALUE, R10                          ;
                                    59 00000000G  00  9E 00009      MOVAB   LIB$SIGNAL, R9                              ;
                                    58 00000000'  00  9E 00010      MOVAB   NETBUF+64, R8                               ;
                                    5E            10  C2 00017      SUBL2   #16, SP                                     ;
                                    08     FA2C   C8  E8 0001A      BLBS    NETUAF_EXISTS, 1$                           ; 1432
                                       00000000G  8F  DD 0001F      PUSHL   #UAF$_NAFDNE                                ; 1433
                                                  5F  11 00025      BRB     3$                                          ;
    0064   8F           20          6E            00  2C 00027 1$: MOVC5   #0, (SP), #32, #100, NETBUF                 ; 1438
                                          C0      A8       0002E
                                       FA1C   C8  9F 00030      PUSHAB  TOKENDSC                                    ; 1443
                                    00000000'  00  9F 00034      PUSHAB  SD_TOKEN1                                   ;
                                    6A            02  FB 0003A      CALLS   #2, CLI$GET_VALUE                           ;
                                    5E            DD  0003D      PUSHL   SP                                          ; 1448
                                          08      AE  9F 0003F      PUSHAB  REMUSER_PTR                                 ;
                                          10      AE  9F 00042      PUSHAB  NODE_LEN                                    ;
                                          18      AE  9F 00045      PUSHAB  NODE_PTR                                    ;
                                00000000V 00            04  FB 00048      CALLS   #4, REMOTE_PARSE                            ;
                                          01            50  E8 0004F      BLBS    R0, 2$                                      ;
                                                        04  00052      RET                                                 ;
    20         20        0C     BE          08      AE  2C 00053 2$:     MOVC5   NODE_LEN, @NODE_PTR, #32, #32, NETBUF       ; 1453
                                          C0      A8       0005A
    20         20        04     BE                  6E  2C 0005C         MOVC5   REMUSER_LEN, @REMUSER_PTR, #32, #32, -      ; 1454
                                          E0      A8       00062                NETBUF+32                                   ;
                                       FA1C   C8  9F 00064      PUSHAB  TOKENDSC                                    ; 1459
                                    00000000'  00  9F 00068      PUSHAB  SD_TOKEN2                                   ;
                                    6A            02  FB 0006E      CALLS   #2, CLI$GET_VALUE                           ;
                                    56     FA1C   C8  3C 00071      MOVZWL  TOKENLEN, LOCUSER_LEN                       ; 1461
                                    57     FA20   C8  D0 00076      MOVL    TOKENPTR, LOCUSER_PTR                       ; 1462
                                    0C            56  D1 0007B      CMPL    LOCUSER_LEN, #12                            ; 1464
                                                  08  1B 0007E      BLEQU   4$                                          ;
                                       00000000G  8F  DD 00080      PUSHL   #UAF$_NAMETOOBIG                            ; 1465
                                                  77  11 00086 3$:  BRB     9$                                          ;
                                    01     FA1C   C8  B1 00088 4$:  CMPW    TOKENLEN, #1                                ; 1470
                                                  14  12 0008D      BNEQ    5$                                          ;
                                    50     FA20   C8  D0 0008F      MOVL    TOKENPTR, R0                                ;
                                    2A            60  91 00094      CMPB    (R0), #42                                   ;
                                                  0A  12 00097      BNEQ    5$                                          ;
```

```
                                56                  6E   D0 00099          MOVL    REMUSER_LEN, LOCUSER_LEN                ; 1473
                            68      E0  A8          20   28 0009C          MOVC3   #32, NETBUF+32, NETBUF+64               ; 1474
                                        06          06   11 000A1          BRB     6$                                      ; 1470
                    20          20      67          56   2C 000A3 5$:      MOVC5   LOCUSER_LEN, (LOCUSER_PTR), #32, #32, -  ; 1481
                                                    68       000A8          NETBUF+64
                                                    7E   D4 000A9 6$:      CLRL    -(SP)                                   ; 1487
                                        0140        8F   BB 000AB          PUSHR   #^M<R6,R8>
                            00000000V   00          03   FB 000AF          CALLS   #3, LOCATE_USER
                                        1A          50   E8 000B6          BLBS    R0, 8$
                                        01          56   D1 000B9          CMPL    LOCUSER_LEN, #1                         ; 1488
                                                    05   12 000BC          BNEQ    7$
                                        2A          68   91 000BE          CMPB    NETBUF+64, #42
                                                    10   13 000C1          BEQL    8$
                                        0140        8F   BB 000C3 7$:      PUSHR   #^M<R6,R8>                              ; 1489
                                                    02   DD 000C7          PUSHL   #2
                                00000000G   8F      DD 000C9          PUSHL   #UAF$_BADUSR
                                        69          04   FB 000CF          CALLS   #4, LIB$SIGNAL
                                                    04   000D2          RET
                                008A    C8      64  8F   9B 000D3 8$:      MOVZBW  #100, NAFRAB+34                         ; 1491
                                            68      A8   9F 000D9          PUSHAB  NAFRAB                                  ; 1496
                            00000000G   00          01   FB 000DC          CALLS   #1, SYS$PUT
                                FA34    C8          50   D0 000E3          MOVL    R0, RMSERR
                                        27          50   E8 000E8          BLBS    R0, 11$
                                        50      FA34    C8   D0 000EB          MOVL    RMSERR, R0                      ; 1499
                            000184EC    8F          50   D1 000F0          CMPL    R0, #99564
                                                    0A   12 000F7          BNEQ    10$
                                00000000G   8F      DD 000F9          PUSHL   #UAF$_NAFUAEERR                         ; 1501
                                        69          01   FB 000FF 9$:      CALLS   #1, LIB$SIGNAL
                                                    04   00102          RET
                                                    50   DD 00103 10$:     PUSHL   R0                                      ; 1503
                                                    7E   D4 00105          CLRL    -(SP)
                                00000000G   8F      DD 00107          PUSHL   #UAF$_NAFADDERR
                                        69          03   FB 0010D          CALLS   #3, LIB$SIGNAL
                                                    16   11 00110          BRB     12$                                     ; 1498
                                00000000G   8F      DD 00112 11$:     PUSHL   #UAF$_NAFADDMSG                         ; 1507
                                        69          01   FB 00118          CALLS   #1, LIB$SIGNAL
                                00010003    8F      DD 0011B          PUSHL   #65539                                  ; 1508
                            00000000V   00          01   FB 00121          CALLS   #1, SECURITY_AUDIT
                            00000000'   00          01   D0 00128 12$:     MOVL    #1, NETUAF_MODIFIED                     ; 1511
                                                    04   0012F          RET                                             ; 1512

; Routine Size:  304 bytes,    Routine Base:  $CODE$ + 050C
```

```
;  1424          1513  1 %sbttl 'remote_parse - parses "node::remoteuser"'
;  1425          1514  1 routine remote_parse (node_ptr, node_len, remuser_ptr, remuser_len) =
;  1426          1515  2 begin
;  1427          1516  2
;  1428          1517  2 !++
;  1429          1518  2 !
;  1430          1519  2 ! FUNCTIONAL DESCRIPTION:
;  1431          1520  2 !
;  1432          1521  2 !       This routine parses a remote user specification in the form
;  1433          1522  2 !          node::remuser, and returns the two components by lengths
;  1434          1523  2 !          and pointers to the strings
;  1435          1524  2 !
;  1436          1525  2 ! INPUTS:
;  1437          1526  2 !
;  1438          1527  2 !       node_ptr - returned as pointer to nodename
;  1439          1528  2 !       node_len -      "       " length of nodename
;  1440          1529  2 !       remuser_ptr - returned as pointer to remote user name
;  1441          1530  2 !       remuser_len -      "       "    length of remote user name
;  1442          1531  2 !
;  1443          1532  2 ! IMPLICIT INPUTS:
;  1444          1533  2 !
;  1445          1534  2 !       TOKENLEN and TOKENPTR - the remote user specification is assumed
;  1446          1535  2 !          to have just been fetched from the command line
;  1447          1536  2 !
;  1448          1537  2 ! OUTPUTS:
;  1449          1538  2 !
;  1450          1539  2 !       none
;  1451          1540  2 !
;  1452          1541  2 ! ROUTINE VALUE:
;  1453          1542  2 !
;  1454          1543  2 !       TRUE if parsed successfully
;  1455          1544  2 !       FALSE if error encountered in parsing
;  1456          1545  2 !
;  1457          1546  2 !--
;  1458          1547  2
;  1459          1548  2 map
;  1460          1549  2         dbl_colon       : vector;
;  1461          1550  2
;  1462          1551  2 local
;  1463          1552  2         dbl_colon_ptr;
;  1464          1553  2
;  1465          1554  2 !
;  1466          1555  2 ! Better be able to find a double colon in the remotename...
;  1467          1556  2 !
;  1468          1557  2 dbl_colon_ptr = ch$find_sub (.tokenlen, .tokenptr, 2, .dbl_colon [1]);
;  1469          1558  2
;  1470          1559  2 if .dbl_colon_ptr eql 0                           ! no double colon found
;  1471          1560  2 or .dbl_colon_ptr eql .tokenptr                   ! no node found
;  1472          1561  3 or .dbl_colon_ptr eql (.tokenptr + .tokenlen - 2) ! no remote user found
;  1473          1562  2 then return LIB$SIGNAL(UAF$_BADNODFORM);
;  1474          1563  2
;  1475          1564  2 !
;  1476          1565  2 ! Determine node length and pointer
;  1477          1566  2 !
;  1478          1567  2 .node_len = .dbl_colon_ptr - .tokenptr;
;  1479          1568  2 .node_ptr = .tokenptr;
;  1480          1569  2
```

```
;  1481          1570  2 !
;  1482          1571  2 ! Make sure node name isn't too long
;  1483          1572  2 !
;  1484          1573  2 !***if . (.node_len) gtru naf$s_node
;  1485          1574  2 if . (.node_len) gtru 6
;  1486          1575  2 then return LIB$SIGNAL(UAF$_NODTOOBIG);
;  1487          1576  2
;  1488          1577  2 !
;  1489          1578  2 ! Determine remote username length and pointer
;  1490          1579  2 !
;  1491          1580  2 .remuser_len = .tokenlen - .(.node_len) - 2;
;  1492          1581  2 .remuser_ptr = .dbl_colon_ptr + 2;
;  1493          1582  2
;  1494          1583  2 !
;  1495          1584  2 ! And make sure name isn't too long
;  1496          1585  2 !
;  1497          1586  2 !***if . (.remuser_len) gtru naf$s_remuser
;  1498          1587  2 if . (.remuser_len) gtru 12
;  1499          1588  2 then return LIB$SIGNAL(UAF$_NAMETOOBIG);
;  1500          1589  2
;  1501          1590  2 return true;
;  1502          1591  1 end;


                                    007C 00000 REMOTE_PARSE:
                                                .WORD   Save R2,R3,R4,R5,R6                     ; 1514
                          56 00000000'  00  9E 00002    MOVAB   TOKENLEN, R6
                          55            66  3C 00009    MOVZWL  TOKENLEN, R5                    ; 1557
                          54      04    A6  D0 0000C    MOVL    TOKENPTR, R4
                          50 00000000'  00  D0 00010    MOVL    DBL_COLON+4, R0
    64          55        60            02  39 00017    MATCHC  #2, (R0), R5, (R4)
                                        03  13 0001C    BEQL    1$
                          53            02  D0 0001E    MOVL    #2, R3
                          53            02  C2 00021 1$: SUBL2   #2, R3
                                        0F  13 00024    BEQL    2$                              ; 1559
                          54            53  D1 00026    CMPL    DBL_COLON_PTR, R4               ; 1560
                                        0A  13 00029    BEQL    2$
                          50      FE A544   9E 0002B    MOVAB   -2(R5)[R4], R0                  ; 1561
                          50            53  D1 00030    CMPL    DBL_COLON_PTR, R0
                                        08  12 00033    BNEQ    3$
                             00000000G  8F  DD 00035 2$: PUSHL   #UAF$_BADNODFORM                ; 1562
                                        38  11 0003B    BRB     5$
                          50      04    A6  D0 0003D 3$: MOVL    TOKENPTR, R0                    ; 1567
          08    BC        53            50  C3 00041    SUBL3   R0, DBL_COLON_PTR, @NODE_LEN
                    04    BC            50  D0 00046    MOVL    R0, @NODE_PTR                   ; 1568
                          06      08    BC  D1 0004A    CMPL    @NODE_LEN, #6                   ; 1574
                                        08  1B 0004E    BLEQU   4$
                             00000000G  8F  DD 00050    PUSHL   #UAF$_NODTOOBIG                 ; 1575
                                        1D  11 00056    BRB     5$
                          50            66  3C 00058 4$: MOVZWL  TOKENLEN, R0                    ; 1580
                          50      08    BC  C2 0005B    SUBL2   @NODE_LEN, R0
                    10    BC      FE    A0  9E 0005F    MOVAB   -2(R0), @REMUSER_LEN
                    0C    BC      02    A3  9E 00064    MOVAB   2(R3), @REMUSER_PTR             ; 1581
                          0C      10    BC  D1 00069    CMPL    @REMUSER_LEN, #12               ; 1587
```

```
                                    0E  1B 0006D          BLEQU   6$                                                      ;
                          00000000G 8F  DD 0006F          PUSHL   #UAF$_NAMETOOBIG                                        ; 1588
                00000000G 00        01  FB 00075 5$:      CALLS   #1, LIB$SIGNAL                                          ;
                                        04 0007C          RET                                                             ;
                          50        01  D0 0007D 6$:      MOVL    #1, R0                                                  ; 1590
                                        04 00080          RET                                                             ; 1591

; Routine Size:  129 bytes,    Routine Base:  $CODE$ + 063C
```

```
;1504    1592  1 %sbttl 'copy_uaf - copy user record'
;1505    1593  1 global routine copy_uaf  =
;1506    1594  2 begin
;1507    1595  2
;1508    1596  2 !++
;1509    1597  2 !
;1510    1598  2 ! FUNCTIONAL DESCRIPTION:
;1511    1599  2 !
;1512    1600  2 !       Routine to copy a user authorization record, giving the
;1513    1601  2 !       new authorization record a different name.
;1514    1602  2 !
;1515    1603  2 ! INPUTS:
;1516    1604  2 !
;1517    1605  2 !       none
;1518    1606  2 !
;1519    1607  2 ! IMPLICIT INPUTS:
;1520    1608  2 !
;1521    1609  2 !       none
;1522    1610  2 !
;1523    1611  2 ! OUTPUTS:
;1524    1612  2 !
;1525    1613  2 !       none
;1526    1614  2 !
;1527    1615  2 ! ROUTINE VALUE:
;1528    1616  2 !
;1529    1617  2 !        false if the copy fails;
;1530    1618  2 !        true  if the copy succeeds.
;1531    1619  2 !
;1532    1620  2 ! SIDE EFFECTS:
;1533    1621  2 !
;1534    1622  2 !       A user record is added.
;1535    1623  2 !
;1536    1624  2 !
;1537    1625  2 !--
;1538    1626  2
;1539    1627  2 local
;1540    1628  2         status,
;1541    1629  2         flag,
;1542    1630  2         lock_rec,
;1543    1631  2         def_sys,
;1544    1632  2         old_user_buffer : vector [uaf$s_username, byte];
;1545    1633  2
;1546    1634  2 map
;1547    1635  2         tokenptr        : ref vector [,byte];
;1548    1636  2
;1549    1637  2 uaf$gl_ctlmsk[uaf$v_copy] = not .uaf$gl_ctlmsk[uaf$v_rename];
;1550    1638  2 !
;1551    1639  2 ! If this is a COPY directly from the UAF> prompt, the authorization
;1552    1640  2 ! record need not be locked, and the default and system records may be
;1553    1641  2 ! copied.  HOWEVER, if this COPY is part of a RENAME operation, the record
;1554    1642  2 ! must be locked, and the default and system records may not be renamed.
;1555    1643  2 ! (The RENAME operation is similar to the COPY operation except that
;1556    1644  2 ! the original record is REMOVE'd.  COPY leaves both records.)
;1557    1645  2 !
;1558    1646  2 if not .uaf$gl_ctlmsk[uaf$v_rename]
;1559    1647  2 then
;1560    1648  3     begin
```

```
;  1561          1649  3     lock_rec = false;                       ! A COPY operation
;  1562          1650  3     def_sys  = true;
;  1563          1651  3     flag = false;
;  1564          1652  3     end
;  1565          1653  2 else
;  1566          1654  3     begin
;  1567          1655  3     lock_rec = true;                        ! A RENAME operation
;  1568          1656  3     def_sys  = false;
;  1569          1657  3     flag = true;
;  1570          1658  2     end;
;  1571          1659  2
;  1572          1660  2 !
;  1573          1661  2 !  Place record to be copied into RECBUF
;  1574          1662  2 !  (If the third argument is true, the call to GET_USER_RECORD
;  1575          1663  2 !  is part of a RENAME operation, and the first token should be saved.
;  1576          1664  2 !  If the third argument is false, the call is part of a COPY
;  1577          1665  2 !  operation, and the first token need not be saved.)
;  1578          1666  2 !
;  1579          1667  2
;  1580          1668  2 if get_user_record (.lock_rec, .def_sys, .flag)
;  1581          1669  2 then                                        ! Make sure a username was
;  1582          1670  3     begin                                   ! specified
;  1583          1671  3     if not cli$present (sd_token2)
;  1584          1672  3     or not cli$get_value (sd_token2, tokendsc)
;  1585          1673  3     or .tokenlen eql 0
;  1586          1674  3     then return LIB$SIGNAL(UAF$_NOUSERNAME);
;  1587          1675  3
;  1588          1676  3 !
;  1589          1677  3 !  Make sure that the new username isn't too long
;  1590          1678  3 !
;  1591          1679  3 !***     if .tokenlen  gtru uaf$s_username
;  1592          1680  3     if .tokenlen  gtru 12
;  1593          1681  3     then LIB$SIGNAL(UAF$_NAMETOOBIG);
;  1594          1682  3
;  1595          1683  3 !
;  1596          1684  3 ! Make sure that the new username is legal
;  1597          1685  3 !
;  1598          1686  3     incru i to .tokenlen - 1
;  1599          1687  3     do
;  1600          1688  3         if ch$fail (ch$find_ch (.symbol_str<0,8>,
;  1601          1689  3                                 symbol_str + 1,
;  1602          1690  3                                .tokenptr [.i]))
;  1603          1691  3         then return LIB$SIGNAL(UAF$_INVUSERNAME);
;  1604          1692  3
;  1605          1693  3 !
;  1606          1694  3 ! If this is a rename save the new user name
;  1607          1695  3 !
;  1608          1696  3 if .uaf$gl_ctlmsk[uaf$v_rename]
;  1609          1697  3 then
;  1610          1698  4     begin
;  1611          1699  4     ch$move (.tokenlen, .tokenptr, newusername);
;  1612          1700  4     newuserlen = .tokenlen;
;  1613          1701  3     end ;
;  1614          1702  3
;  1615          1703  3 !
;  1616          1704  3 ! Place the new username in RECBUF, but save the old username first
;  1617          1705  3 !
```

```
; 1618          1706  3     ch$move (uaf$s_username, recbuf[uaf$t_username], old_user_buffer);
; 1619          1707  3     ch$copy (.tokenlen, .tokenptr, ' ',
; 1620          1708  3             uaf$s_username, recbuf[uaf$t_username]);
; 1621          1709  3     pwd_flag = true;
; 1622          1710  3
; 1623          1711  3     status = update_record ();
; 1624          1712  3     if not .status
; 1625          1713  3     then return false;
; 1626          1714  3
; 1627          1715  3 !
; 1628          1716  3 ! If this is a copy operation then zero fill the last login info
; 1629          1717  3 !
; 1630          1718  3 if not .uaf$gl_ctlmsk[uaf$v_rename]
; 1631          1719  3 then
; 1632          1720  4     begin
; 1633          1721  4     recbuf[uaf$w_logfails] = 0 ;
; 1634          1722  4     ch$fill ( 0, uaf$s_lastlogin_i, recbuf[uaf$q_lastlogin_i] ) ;
; 1635          1723  4     ch$fill ( 0, uaf$s_lastlogin_n, recbuf[uaf$q_lastlogin_n] ) ;
; 1636          1724  3     end ;
; 1637          1725  3
; 1638          1726  3 !
; 1639          1727  3 ! Now output the new record
; 1640          1728  3 !
; 1641          1729  4     if rmsbad ($put (rab = uafrab))
; 1642          1730  3     then
; 1643          1731  4         begin
; 1644          1732  4         if .rmserr eql rms$_dup
; 1645          1733  4         then
; 1646          1734  4             return LIB$SIGNAL(UAF$_UAEERR)      ! username already exists
; 1647          1735  4         else
; 1648          1736  5             begin
; 1649          1737  5             LIB$SIGNAL(UAF$_ADDERR, 0, .rmserr);
; 1650          1738  5             return false;
; 1651          1739  5             end
; 1652          1740  4         end
; 1653          1741  3     else
; 1654          1742  3 !
; 1655          1743  3 ! The copy was successful -- tell the user and set modify flag
; 1656          1744  3 !
; 1657          1745  4         begin
; 1658          1746  4         modify_flag = true;
; 1659          1747  5         security_audit ((if not .uaf$gl_ctlmsk[uaf$v_rename]
; 1660          1748  5                          then nsa$k_recid_sysuaf_cop
; 1661          1749  4                          else nsa$k_recid_sysuaf_ren),
; 1662          1750  4                         old_user_buffer);
; 1663          1751  4         if not .uaf$gl_ctlmsk[uaf$v_rename]
; 1664          1752  4         then
; 1665          1753  5             begin
; 1666          1754  5             LIB$SIGNAL(UAF$_COPMSG);
; 1667          1755  6             if (.uaf$gl_ctlmsk[uaf$v_cli]
; 1668          1756  6             and not .uaf$gl_ctlmsk[uaf$v_clitables])
; 1669          1757  5             and .uaf$gl_ctlmsk[uaf$v_clitab_pres]
; 1670          1758  5             then LIB$SIGNAL(UAF$_CLIWARNMSG);
; 1671          1759  5             !
; 1672          1760  5             ! Since passwords are folded in with the username, passwords for
; 1673          1761  5             ! COPYed records will no longer work--warn the user
; 1674          1762  5             !
```

```
;  1675          1763  5             if .pwd_flag
;  1676          1764  5             then LIB$SIGNAL(UAF$_DEFPWD);
;  1677          1765  5             uaf$gl_ctlmsk[uaf$v_copy] = false;
;  1678          1766  6             if (cli$present ( sd_add_identifier ) and
;  1679          1767  6                 .rdb_exists )
;  1680          1768  5             then
;  1681          1769  6                 begin
;  1682          1770  6                 !
;  1683          1771  6                 ! Set the resource attribute if /ATTRIB=RESOURCE was specified
;  1684          1772  6                 ! and then add the appropriate identifiers to the rights data base
;  1685          1773  6                 !
;  1686          1774  6                 if cli$present (sd_attribresource)
;  1687          1775  6                     then attributes = kgb$m_resource
;  1688          1776  6                     else attributes = 0 ;
;  1689          1777  6                 uaf$add_ident_recbuf () ;
;  1690          1778  5                 end;
;  1691          1779  4             end;
;  1692          1780  3         end ;
;  1693          1781  3     end
;  1694          1782  3
;  1695          1783  3 !
;  1696          1784  3 ! The attempt to GET_USER_RECORD failed...
;  1697          1785  3 !
;  1698          1786  2 else return false;
;  1699          1787  2
;  1700          1788  2 !
;  1701          1789  2 ! If we get here, everything succeeded -- return true
;  1702          1790  2 !
;  1703          1791  2 return true;
;  1704          1792  2
;  1705          1793  1 end;


                                        0FFC 00000          .ENTRY  COPY_UAF, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-  ; 1593
                                                                    R11                                           ;
                                5B 00000000G    00  9E 00002        MOVAB   CLI$PRESENT, R11                      ;
                                5A 00000000'    00  9E 00009        MOVAB   NEWUSERNAME, R10                      ;
                                59 00000000'    00  9E 00010        MOVAB   SD_TOKEN2, R9                         ;
                                58 00000000G    00  9E 00017        MOVAB   LIB$SIGNAL, R8                        ;
                                57 00000000'    00  9E 0001E        MOVAB   UAF$GL_CTLMSK, R7                     ;
                                5E                  20  C2 00025    SUBL2   #32, SP                               ;
50              67              01                  01  EF 00028    EXTZV   #0, #1, UAF$GL_CTLMSK, R0             ; 1637
                                50                  50  92 0002D    MCOMB   R0, R0                                ;
67              01              02                  50  F0 00030    INSV    R0, #2, #1, UAF$GL_CTLMSK             ;
                                07                  67  E8 00035    BLBS    UAF$GL_CTLMSK, 1$                     ; 1646
                                51                  01  7D 00038    MOVQ    #1, DEF_SYS                           ; 1650
                                                    50  D4 0003B    CLRL    FLAG                                  ; 1651
                                                    06  11 0003D    BRB     2$                                    ; 1646
                                52                  01  D0 0003F 1$: MOVL    #1, LOCK_REC                          ; 1655
                                50                  01  7D 00042    MOVQ    #1, FLAG                              ; 1657
                                                    50  DD 00045 2$: PUSHL   FLAG                                  ; 1668
                                                    51  DD 00047    PUSHL   DEF_SYS                               ;
                                                    52  DD 00049    PUSHL   LOCK_REC                              ;
                    00000000V   00                  03  FB 0004B    CALLS   #3, GET_USER_RECORD                   ;
```

```
                         03              50  E8 00052          BLBS    R0, 3$                                          ;
                                       0169  31 00055          BRW     25$                                             ;
                                         59  DD 00058 3$:      PUSHL   R9                                              ; 1671
                         6B              01  FB 0005A          CALLS   #1, CLI$PRESENT                                 ;
                         14              50  E9 0005D          BLBC    R0, 4$                                          ;
                                   5C    A7  9F 00060          PUSHAB  TOKENDSC                                        ; 1672
                                         59  DD 00063          PUSHL   R9                                              ;
                   00000000G     00        02  FB 00065          CALLS   #2, CLI$GET_VALUE                               ;
                         05              50  E9 0006C          BLBC    R0, 4$                                          ;
                                   5C    A7  B5 0006F          TSTW    TOKENLEN                                        ; 1673
                                         08  12 00072          BNEQ    5$                                              ;
                              00000000G  8F  DD 00074 4$:       PUSHL   #UAF$_NOUSERNAME                                ; 1674
                                         37  11 0007A          BRB     9$                                              ;
                         0C        5C    A7  B1 0007C 5$:      CMPW    TOKENLEN, #12                                   ; 1680
                                         09  1B 00080          BLEQU   6$                                              ;
                              00000000G  8F  DD 00082          PUSHL   #UAF$_NAMETOOBIG                                ; 1681
                         68              01  FB 00088          CALLS   #1, LIB$SIGNAL                                  ;
                         53        5C    A7  3C 0008B 6$:      MOVZWL  TOKENLEN, R3                                    ; 1686
                                         53  D7 0008F          DECL    R3                                              ;
                                         52  D4 00091          CLRL    I                                               ;
                                         22  11 00093          BRB     11$                                             ;
                         51        01DC  C9  9A 00095 7$:      MOVZBL  SYMBOL_STR, R1                                  ; 1688
                         50        60    A7  D0 0009A          MOVL    TOKENPTR, R0                                    ; 1690
                01DD     C9              51        6240  3A 0009E          LOCC    (I)[R0], R1, SYMBOL_STR+1                       ;
                                         02  12 000A5          BNEQ    8$                                              ;
                                         51  D4 000A7          CLRL    R1                                              ;
                                         51  D5 000A9 8$:      TSTL    R1                                              ;
                                         08  12 000AB          BNEQ    10$                                             ;
                              00000000G  8F  DD 000AD          PUSHL   #UAF$_INVUSERNAME                               ; 1691
                                         78  11 000B3 9$:      BRB     14$                                             ;
                                         52  D6 000B5 10$:     INCL    I                                               ; 1690
                              53         52  D1 000B7 11$:     CMPL    I, R3                                           ;
                                         D9  1B 000BA          BLEQU   7$                                              ;
                         10              67  E9 000BC          BLBC    UAF$GL_CTLMSK, 12$                              ; 1696
                         56        5C    A7  3C 000BF          MOVZWL  TOKENLEN, R6                                    ; 1699
                         50        60    A7  D0 000C3          MOVL    TOKENPTR, R0                                    ;
                   6A            60        56  28 000C7          MOVC3   R6, (R0), NEWUSERNAME                           ;
                         FC        AA        56  D0 000CB          MOVL    R6, NEWUSERLEN                                  ; 1700
                   6E      0080    C7        20  28 000CF 12$:     MOVC3   #32, RECBUF+4, OLD_USER_BUFFER                  ; 1706
                                   50  60    A7  D0 000D5          MOVL    TOKENPTR, R0                                    ; 1707
                20 20               60  5C    A7  2C 000D9          MOVC5   TOKENLEN, (R0), #32, #32, RECBUF+4              ; 1708
                                         0080  C7       000DF                                                          ;
                              06F4       C7        01  D0 000E2          MOVL    #1, PWD_FLAG                            ; 1709
                              00000000G  00        00  FB 000E7          CALLS   #0, UPDATE_RECORD                       ; 1711
                                         4D        50  E9 000EE          BLBC    STATUS, 16$                             ; 1712
                                         14        67  E8 000F1          BLBS    UAF$GL_CTLMSK, 13$                      ; 1718
                                               01E0  C7  B4 000F4          CLRW    RECBUF+356                              ; 1721
                08 00                  6E             00  2C 000F8          MOVC5   #0, (SP), #0, #8, RECBUF+396            ; 1722
                                               0208  C7       000FD                                                          ;
                08 00                  6E             00  2C 00100          MOVC5   #0, (SP), #0, #8, RECBUF+404            ; 1723
                                               0210  C7       00105                                                          ;
                                               0664  C7  9F 00108 13$:     PUSHAB  UAFRAB                                  ; 1729
                              00000000G  00        01  FB 0010C          CALLS   #1, SYS$PUT                             ;
                                   74    A7        50  D0 00113          MOVL    R0, RMSERR                              ;
                                         27        50  E8 00117          BLBS    R0, 17$                                 ;
                                         50  74    A7  D0 0011A          MOVL    RMSERR, R0                              ; 1732
                              000184EC   8F        50  D1 0011E          CMPL    R0, #99564                              ;
```

```
                                    0A  12 00125           BNEQ    15$                                             ;
                          00000000G 8F  DD 00127           PUSHL   #UAF$_UAEERR                                    ; 1734
                         68         01  FB 0012D 14$:     CALLS   #1, LIB$SIGNAL                                  ;
                                    04  00130           RET                                                     ; 1736
                                    50  DD 00131 15$:     PUSHL   R0                                              ; 1737
                                    7E  D4 00133           CLRL    -(SP)                                           ;
                          00000000G 8F  DD 00135           PUSHL   #UAF$_ADDERR                                    ;
                         68         03  FB 0013B           CALLS   #3, LIB$SIGNAL                                  ;
                               0080 31  00013E 16$:        BRW     25$                                             ; 1738
                      CC   AA       01  D0 00141 17$:     MOVL    #1, MODIFY_FLAG                                 ; 1746
                                    5E  DD 00145           PUSHL   SP                                              ; 1747
                         08         67  E8 00147           BLBS    UAF$GL_CTLMSK, 18$                              ;
                            00040002 8F  DD 0014A           PUSHL   #262146                                         ;
                                    06  11 00150           BRB     19$                                             ;
                            00050002 8F  DD 00152 18$:     PUSHL   #327682                                         ;
                 00000000V  00       02  FB 00158 19$:     CALLS   #2, SECURITY_AUDIT                              ;
                         5B         67  E8 0015F           BLBS    UAF$GL_CTLMSK, 24$                              ; 1751
                          00000000G 8F  DD 00162           PUSHL   #UAF$_COPMSG                                    ; 1754
                         68         01  FB 00168           CALLS   #1, LIB$SIGNAL                                  ;
                   11    67         03  E1 0016B           BBC     #3, UAF$GL_CTLMSK, 20$                          ; 1755
                   0D    67         04  E0 0016F           BBS     #4, UAF$GL_CTLMSK, 20$                          ; 1756
                   09    67         05  E1 00173           BBC     #5, UAF$GL_CTLMSK, 20$                          ; 1757
                          00000000G 8F  DD 00177           PUSHL   #UAF$_CLIWARNMSG                                ; 1758
                         68         01  FB 0017D           CALLS   #1, LIB$SIGNAL                                  ;
                         09     06F4 C7  E9 00180 20$:     BLBC    PWD_FLAG, 21$                                   ; 1763
                          00000000G 8F  DD 00185           PUSHL   #UAF$_DEFPWD                                    ; 1764
                         68         01  FB 0018B           CALLS   #1, LIB$SIGNAL                                  ;
                         67         04  8A 0018E 21$:     BICB2   #4, UAF$GL_CTLMSK                               ; 1765
                               34   A9  9F 00191           PUSHAB  SD_ADD_IDENTIFIER                               ; 1766
                         6B         01  FB 00194           CALLS   #1, CLI$PRESENT                                 ;
                         23         50  E9 00197           BLBC    R0, 24$                                         ;
                         1F     70  A7  E9 0019A           BLBC    RDB_EXISTS, 24$                                 ; 1767
                               80   AA  9F 0019E           PUSHAB  SD_ATTRIBRESOURCE                               ; 1774
                         6B         01  FB 001A1           CALLS   #1, CLI$PRESENT                                 ;
                         09         50  E9 001A4           BLBC    R0, 22$                                         ;
                 00000000G  00      01  D0 001A7           MOVL    #1, ATTRIBUTES                                  ; 1775
                                    06  11 001AE           BRB     23$                                             ;
                          00000000G 00  D4 001B0 22$:     CLRL    ATTRIBUTES                                      ; 1776
                 00000000G  00      00  FB 001B6 23$:     CALLS   #0, UAF$ADD_IDENT_RECBUF                        ; 1777
                         50         01  D0 001BD 24$:     MOVL    #1, R0                                          ; 1791
                                    04  001C0           RET                                                     ;
                                    50  D4 001C1 25$:     CLRL    R0                                              ; 1793
                                    04  001C3           RET                                                     ;

; Routine Size:  452 bytes,    Routine Base:  $CODE$ + 068D
```

```
;  1707          1794  1 %sbttl 'create_proxy - create NETUAF.DAT proxy file'
;  1708          1795  1 global routine create_proxy : novalue =
;  1709          1796  2 begin
;  1710          1797  2
;  1711          1798  2 !++
;  1712          1799  2 !
;  1713          1800  2 ! FUNCTIONAL DESCRIPTION:
;  1714          1801  2 !
;  1715          1802  2 !       This routine will create a DECnet Proxy Login File,
;  1716          1803  2 !       if and only if it does not already exist,
;  1717          1804  2 !       called NETUAF.DAT, in order to map remote users into
;  1718          1805  2 !       local accounts.
;  1719          1806  2 !
;  1720          1807  2 ! INPUTS:
;  1721          1808  2 !
;  1722          1809  2 !    none
;  1723          1810  2 !
;  1724          1811  2 ! OUTPUTS:
;  1725          1812  2 !
;  1726          1813  2 !    none
;  1727          1814  2 !
;  1728          1815  2 ! IMPLICIT INPUTS:
;  1729          1816  2 !
;  1730          1817  2 !       none
;  1731          1818  2 !
;  1732          1819  2 ! IMPLICIT OUTPUTS:
;  1733          1820  2 !
;  1734          1821  2 !       none
;  1735          1822  2 !
;  1736          1823  2 ! SIDE EFFECTS:
;  1737          1824  2 !
;  1738          1825  2 !       NETUAF.DAT is created and initialized
;  1739          1826  2 !
;  1740          1827  2 !--
;  1741          1828  2
;  1742          1829  2
;  1743          1830  2 !
;  1744          1831  2 ! NETUAF.DAT should not already exist
;  1745          1832  2 !
;  1746          1833  2 if .netuaf_exists
;  1747          1834  2 then
;  1748          1835  3     begin
;  1749          1836  3     LIB$SIGNAL(UAF$_NAFAEX);
;  1750          1837  3     return;
;  1751          1838  2     end;
;  1752          1839  2
;  1753          1840  2 !
;  1754          1841  2 ! Should be able to create NETUAF.DAT with no problems
;  1755          1842  2 !
;  1756          1843  3 if rmsbad ($create (fab = naffab))
;  1757          1844  2 then LIB$SIGNAL(UAF$_NAFCREERR, 0, .rmserr);
;  1758          1845  2
;  1759          1846  2 !
;  1760          1847  2 ! Should connect ok, too
;  1761          1848  2 !
;  1762          1849  3 if rmsbad ($connect (rab = nafrab))
;  1763          1850  2 then LIB$SIGNAL(UAF$_NAFCONERR, 0, .rmserr);
```

```
;  1764          1851  2
;  1765          1852  2 !
;  1766          1853  2 ! Normal access is by key
;  1767          1854  2 !
;  1768          1855  2 nafrab [rab$b_rac] = rab$c_key;
;  1769          1856  2
;  1770          1857  2 !
;  1771          1858  2 ! Establish proper addresses
;  1772          1859  2 !
;  1773          1860  2 nafrab [rab$l_ubf] = netbuf;
;  1774          1861  2 nafrab [rab$l_rbf] = netbuf;
;  1775          1862  2
;  1776          1863  2 !
;  1777          1864  2 ! Set NETUAF.DAT existence flag
;  1778          1865  2 !
;  1779          1866  2 netuaf_exists = true;
;  1780          1867  2 netuaf_modified = true;
;  1781          1868  2
;  1782          1869  1 end;


                                        000C 00000          .ENTRY  CREATE_PROXY, Save R2,R3                        ; 1795
                              53 00000000G   00   9E 00002  MOVAB   LIB$SIGNAL, R3                                  ;
                              52 00000000'   00   9E 00009  MOVAB   RMSERR, R2                                       ;
                              0A       F8    A2   E9 00010  BLBC    NETUAF_EXISTS, 1$                               ; 1833
                                 00000000G   8F   DD 00014  PUSHL   #UAF$_NAFAEX                                    ; 1836
                              63             01   FB 0001A  CALLS   #1, LIB$SIGNAL                                  ;
                                             04   0001D  RET                                                        ; 1835
                                 00000000'   00   9F 0001E  1$:     PUSHAB  NAFFAB                                  ; 1843
                        00000000G   00       01   FB 00024  CALLS   #1, SYS$CREATE                                 ;
                                    62       50   D0 0002B  MOVL    R0, RMSERR                                      ;
                                    0D       50   E8 0002E  BLBS    R0, 2$                                          ;
                                             62   DD 00031  PUSHL   RMSERR                                         ; 1844
                                             7E   D4 00033  CLRL    -(SP)                                           ;
                                 00000000G   8F   DD 00035  PUSHL   #UAF$_NAFCREERR                                 ;
                                    63       03   FB 0003B  CALLS   #3, LIB$SIGNAL                                  ;
                                       0634  C2   9F 0003E  2$:     PUSHAB  NAFRAB                                  ; 1849
                        00000000G   00       01   FB 00042  CALLS   #1, SYS$CONNECT                                ;
                                    62       50   D0 00049  MOVL    R0, RMSERR                                      ;
                                    0D       50   E8 0004C  BLBS    R0, 3$                                          ;
                                             62   DD 0004F  PUSHL   RMSERR                                         ; 1850
                                             7E   D4 00051  CLRL    -(SP)                                           ;
                                 00000000G   8F   DD 00053  PUSHL   #UAF$_NAFCONERR                                 ;
                                    63       03   FB 00059  CALLS   #3, LIB$SIGNAL                                  ;
                              0652  C2       01   90 0005C  3$:     MOVB    #1, NAFRAB+30                           ; 1855
                              0658  C2   058C  C2   9E 00061  MOVAB   NETBUF, NAFRAB+36                             ; 1860
                              065C  C2   058C  C2   9E 00068  MOVAB   NETBUF, NAFRAB+40                             ; 1861
                                 F8    A2       01   D0 0006F  MOVL    #1, NETUAF_EXISTS                           ; 1866
                        00000000'   00       01   D0 00073  MOVL    #1, NETUAF_MODIFIED                         ; 1867
                                             04   0007A  RET                                                        ; 1869

; Routine Size:  123 bytes,    Routine Base:  $CODE$ + 0881
```

```
;1784        1870  1 %sbttl 'modify_uaf - update user record (s)'
;1785        1871  1 global routine modify_uaf : novalue =
;1786        1872  2 begin
;1787        1873  2
;1788        1874  2 !++
;1789        1875  2 !
;1790        1876  2 ! FUNCTIONAL DESCRIPTION:
;1791        1877  2 !
;1792        1878  2 !       Routine to modify any of the fields in one or more user records.
;1793        1879  2 !
;1794        1880  2 ! INPUTS:
;1795        1881  2 !
;1796        1882  2 !       none
;1797        1883  2 !
;1798        1884  2 ! IMPLICIT INPUTS:
;1799        1885  2 !
;1800        1886  2 !       none
;1801        1887  2 !
;1802        1888  2 ! OUTPUTS:
;1803        1889  2 !
;1804        1890  2 !       none
;1805        1891  2 !
;1806        1892  2 ! IMPLICIT OUTPUTS:
;1807        1893  2 !
;1808        1894  2 !       none
;1809        1895  2 !
;1810        1896  2 ! SIDE EFFECTS:
;1811        1897  2 !
;1812        1898  2 !       none
;1813        1899  2 !
;1814        1900  2 ! ROUTINE VALUE:
;1815        1901  2 !
;1816        1902  2 !       none
;1817        1903  2 !--
;1818        1904  2
;1819        1905  2 local
;1820        1906  2         status;
;1821        1907  2
;1822        1908  2 !
;1823        1909  2 ! Obtain the user specification.  This sets wildcard flags and initializes
;1824        1910  2 ! the appropriate key in RECBUF.
;1825        1911  2 !
;1826        1912  2
;1827        1913  2 if not parse_wild (sd_token1,false)              ! Null string is disallowed
;1828        1914  2 then return;
;1829        1915  2
;1830        1916  2 uaf$rab[rab$l_rop] = rab$m_rlk;           ! Lock records for writing
;1831        1917  2 rabptr = outrab;
;1832        1918  2 found_match = false;
;1833        1919  2
;1834        1920  3 if rmsbad (status = wild_user (modify_rec))      ! Modify each record
;1835        1921  2 then
;1836        1922  3     begin
;1837        1923  3     if .rmserr eql rms$_rnf
;1838        1924  3     then
;1839        1925  3         LIB$SIGNAL(UAF$_BADSPC)
;1840        1926  3     else
```

```
;  1841          1927  4          (if .rmserr neq 0
;  1842          1928  4           then
;  1843          1929  4               LIB$SIGNAL(UAF$_MDFYERR, 0, .rmserr))
;  1844          1930  3      end
;  1845          1931  2  else
;  1846          1932  3      begin
;  1847          1933  3      if .status and .found_match
;  1848          1934  3      then
;  1849          1935  4          begin
;  1850          1936  4          LIB$SIGNAL(UAF$_MDFYMSG);
;  1851          1937  5          if (.uaf$gl_ctlmsk[uaf$v_cli]
;  1852          1938  5          and not .uaf$gl_ctlmsk[uaf$v_clitables])
;  1853          1939  4          and .uaf$gl_ctlmsk[uaf$v_clitab_pres]
;  1854          1940  4          then LIB$SIGNAL(UAF$_CLIWARNMSG);
;  1855          1941  3          end;
;  1856          1942  2      end;
;  1857          1943  1  end;


                                                001C 00000          .ENTRY  MODIFY_UAF, Save R2,R3,R4                       ; 1871
                                    54 00000000G   00   9E 00002    MOVAB   LIB$SIGNAL, R4                                  ;
                                    53 00000000'   00   9E 00009    MOVAB   UAF$GL_CTLMSK, R3                               ;
                                                   7E   D4 00010    CLRL    -(SP)                                           ; 1913
                                       00000000'   00   9F 00012    PUSHAB  SD_TOKEN1                                       ;
                                00000000G  00              02   FB 00018    CALLS   #2, PARSE_WILD                          ;
                                           71              50   E9 0001F    BLBC    R0, 4$                                  ;
                                    0668   C3 00080000     8F   D0 00022    MOVL    #524288, UAFRAB+4                       ; 1916
                                      F4   A3      06F8     C3   9E 0002B    MOVAB   OUTRAB, RABPTR                          ; 1917
                                           A3      073C     C3   D4 00031    CLRL    FOUND_MATCH                             ; 1918
                                       00000000V   00   9F 00035    PUSHAB  MODIFY_REC                                      ; 1920
                                00000000V  00              01   FB 0003B    CALLS   #1, WILD_USER                           ;
                                          74   A3          50   D0 00042    MOVL    STATUS, RMSERR                          ;
                                               27          50   E8 00046    BLBS    STATUS, 2$                              ;
                                               52      74  A3   D0 00049    MOVL    RMSERR, R2                              ; 1923
                                00018282       8F              52   D1 0004D    CMPL    R2, #98994                      ;
                                                               08   12 00054    BNEQ    1$                              ;
                                       00000000G   8F          DD   00056    PUSHL   #UAF$_BADSPC                    ; 1925
                                                               32   11 0005C    BRB     3$                              ;
                                                               52   D5 0005E 1$: TSTL    R2                              ; 1927
                                                               31   13 00060    BEQL    4$                              ;
                                                               52   DD 00062    PUSHL   R2                              ; 1929
                                                               7E   D4 00064    CLRL    -(SP)                           ;
                                       00000000G   8F          DD   00066    PUSHL   #UAF$_MDFYERR                   ;
                                                64             03   FB 0006C    CALLS   #3, LIB$SIGNAL                  ;
                                                               04   00006F    RET                                     ; 1922
                                                1E      073C   C3   E9 00070 2$: BLBC    FOUND_MATCH, 4$                 ; 1933
                                       00000000G   8F          DD   00075    PUSHL   #UAF$_MDFYMSG                   ; 1936
                                                64             01   FB 0007B    CALLS   #1, LIB$SIGNAL                  ;
                                    11          63             03   E1 0007E    BBC     #3, UAF$GL_CTLMSK, 4$           ; 1937
                                    0D          63             04   E0 00082    BBS     #4, UAF$GL_CTLMSK, 4$           ; 1938
                                    09          63             05   E1 00086    BBC     #5, UAF$GL_CTLMSK, 4$           ; 1939
                                       00000000G   8F          DD   0008A    PUSHL   #UAF$_CLIWARNMSG                ; 1940
                                                64             01   FB 00090 3$: CALLS   #1, LIB$SIGNAL                  ;
                                                               04   00093 4$: RET                                     ; 1943
```

; Routine Size: 148 bytes,   Routine Base: $CODE$ + 08FC

```
;1859        1944  1 %sbttl 'modify_rec - update a user record action routine'
;1860        1945  1 routine modify_rec =
;1861        1946  2 begin
;1862        1947  2
;1863        1948  2 !++
;1864        1949  2 !
;1865        1950  2 ! FUNCTIONAL DESCRIPTION:
;1866        1951  2 !
;1867        1952  2 !       Modify an individual user record.
;1868        1953  2 !
;1869        1954  2 ! INPUTS:
;1870        1955  2 !
;1871        1956  2 !       none
;1872        1957  2 !
;1873        1958  2 ! IMPLICIT INPUTS:
;1874        1959  2 !
;1875        1960  2 !       RABPTR - RMS data structure for the file
;1876        1961  2 !
;1877        1962  2 ! OUTPUTS:
;1878        1963  2 !
;1879        1964  2 !       none
;1880        1965  2 !
;1881        1966  2 ! IMPLICIT OUTPUTS:
;1882        1967  2 !
;1883        1968  2 !       none
;1884        1969  2 !
;1885        1970  2 ! SIDE EFFECTS:
;1886        1971  2 !
;1887        1972  2 !       none
;1888        1973  2 !
;1889        1974  2 ! ROUTINE VALUE:
;1890        1975  2 !
;1891        1976  2 !       If an error is encountered the appropriate status is returned
;1892        1977  2 !       except if the error message has already been output in which
;1893        1978  2 !       case 0 is returned.
;1894        1979  2 !--
;1895        1980  2
;1896        1981  2 local
;1897        1982  2     old_uic     : $bblock[4],
;1898        1983  2     new_uic     : $bblock[4],
;1899        1984  2     oldaccname  : vector [uaf$s_account,byte],
;1900        1985  2     newaccname  : vector [uaf$s_account,byte],
;1901        1986  2     oldaccdesc  : statdesc ,
;1902        1987  2     newaccdesc  : statdesc ,
;1903        1988  2     status      : long ;
;1904        1989  2
;1905        1990  2 !
;1906        1991  2 ! User record has been read into RECBUF by caller.  Update values
;1907        1992  2 ! and modify the record.
;1908        1993  2 !
;1909        1994  2 ! When accessing records by uic, this routine is called repeatedly
;1910        1995  2 ! from WILD_USER. UPDATE_RECORD is called to modify the appropriate
;1911        1996  2 ! record fields for each requested record, and therefore must
;1912        1997  2 ! reparse the command line each time.  If call_count is greater
;1913        1998  2 ! than 0, the command line is reparsed.
;1914        1999  2 !
;1915        2000  2
```

```
;  1916     2001  2 !IF .by_account
;  1917     2002  2 !THEN
;  1918     2003  2 !    (IF NOT fmg$match_name (NAMELEN (UAF$S_ACCOUNT,UAF$T_ACCOUNT),
;  1919     2004  2 !                                    RECBUF[UAF$T_ACCOUNT],
;  1920     2005  2 !                                   .match_tokenlen, match_token)
;  1921     2006  2 !     THEN
;  1922     2007  2 !       RETURN TRUE)
;  1923     2008  2 !ELSE
;  1924     2009  2 if .str_wild
;  1925     2010  2 and not fmg$match_name (namelen (uaf$s_username,recbuf[uaf$t_username]),
;  1926     2011  2                          recbuf[uaf$t_username],
;  1927     2012  2                          .match_tokenlen,  match_token)
;  1928     2013  2 then return true;
;  1929     2014  2 found_match = true;
;  1930     2015  2
;  1931     2016  2 if ch$eql (.defuser<0,8>, defuser+1, .tokenlen, .tokenptr, ' ')
;  1932     2017  2 or ch$eql (.defuser<0,8>, defuser+1,
;  1933     2018  2            namelen (uaf$s_username, recbuf[uaf$t_username]),
;  1934     2019  2            recbuf[uaf$t_username], ' ')
;  1935     2020  2 then
;  1936     2021  3     begin
;  1937     2022  3     mod_default = true;
;  1938     2023  3     default_uaf ();
;  1939     2024  3     call_count = .call_count + 1;
;  1940     2025  3     return true;
;  1941     2026  2     end;
;  1942     2027  2
;  1943     2028  2 !
;  1944     2029  2 ! Save the old UIC and account name
;  1945     2030  2 !
;  1946     2031  2 old_uic[uic$v_format] = 0 ;
;  1947     2032  2 old_uic[uic$v_group ] = .recbuf [uaf$w_grp] ;
;  1948     2033  2 old_uic[uic$v_member] = .recbuf [uaf$w_mem] ;
;  1949     2034  2 ch$move ( uaf$s_account, recbuf[uaf$t_account], oldaccname ) ;
;  1950     2035  2 oldaccdesc [length]  = namelen ( uaf$s_account, recbuf[uaf$t_account]) ;
;  1951     2036  2 oldaccdesc [pointer] = oldaccname ;
;  1952     2037  2
;  1953     2038  2 if update_record ()
;  1954     2039  2 then
;  1955     2040  3     begin
;  1956     2041  3     !
;  1957     2042  3     ! Save the new UIC and account name
;  1958     2043  3     !
;  1959     2044  3     new_uic[uic$v_format] = 0 ;
;  1960     2045  3     new_uic[uic$v_group ] = .recbuf [uaf$w_grp] ;
;  1961     2046  3     new_uic[uic$v_member] = .recbuf [uaf$w_mem] ;
;  1962     2047  3     ch$move ( uaf$s_account, recbuf[uaf$t_account], newaccname ) ;
;  1963     2048  3     newaccdesc [length]  = namelen ( uaf$s_account, recbuf[uaf$t_account]) ;
;  1964     2049  3     newaccdesc [pointer] = newaccname ;
;  1965     2050  3
;  1966     2051  3     !
;  1967     2052  3     ! Update the UAF record
;  1968     2053  3     !
;  1969     2054  4     if rmsbad ($update (rab = uafrab))
;  1970     2055  3     then
;  1971     2056  4         begin
;  1972     2057  4         LIB$SIGNAL(UAF$_MDFYERR, 0, .rmserr);
```

```
;  1973          2058  4          return .rmserr
;  1974          2059  4          end
;  1975          2060  3      else
;  1976          2061  4          begin
;  1977          2062  4          modify_flag = true;
;  1978          2063  4          security_audit (nsa$k_recid_sysuaf_mod);
;  1979          2064  4          call_count = .call_count + 1;
;  1980          2065  3          end;
;  1981          2066  3
;  1982          2067  4      if (cli$present ( sd_modify_identifier ) and
;  1983          2068  4          .rdb_exists )
;  1984          2069  3      then
;  1985          2070  4          begin
;  1986          2071  4
;  1987          2072  4          !
;  1988          2073  4          ! If the UIC changed then modify the identifiers
;  1989          2074  4          !
;  1990          2075  4          if .old_uic neq .new_uic
;  1991          2076  4          then
;  1992          2077  5              begin
;  1993          2078  5              local
;  1994          2079  5                  oldidname       : vector [kgb$s_name, byte],
;  1995          2080  5                  oldiddesc       : statdesc ;
;  1996          2081  5
;  1997          2082  5              oldiddesc[length]  = kgb$s_name ;
;  1998          2083  5              oldiddesc[pointer] = oldidname ;
;  1999        P 2084  5              status = $idtoasc ( id     = .old_uic,
;  2000        P 2085  5                                  namlen = oldiddesc[length],
;  2001          2086  5                                  nambuf = oldiddesc ) ;
;  2002          2087  5
;  2003          2088  5              if not .status
;  2004          2089  5              then LIB$SIGNAL(UAF$_RDBMDFYERRU, 2,
;  2005          2090  5                                  .old_uic[uic$v_group],
;  2006          2091  5                                  .old_uic[uic$v_member], .status)
;  2007          2092  5              else if .status
;  2008          2093  5                  then
;  2009          2094  6                      begin
;  2010        P 2095  6                      status = $mod_ident ( id        = .old_uic,
;  2011          2096  6                                            new_value = .new_uic ) ;
;  2012          2097  6
;  2013          2098  6                      if not .status
;  2014          2099  6                          then LIB$SIGNAL(UAF$_RDBMDFYERR, 1, oldiddesc, .status)
;  2015          2100  6                          else
;  2016          2101  7                              begin
;  2017          2102  7                              rightslist_modified = true ;
;  2018          2103  7                              LIB$SIGNAL(UAF$_RDBMDFYMSG, 1, oldiddesc);
;  2019          2104  6                              end ;
;  2020          2105  5                      end ;
;  2021          2106  4              end;
;  2022          2107  3          end ;
;  2023          2108  3      return true ;
;  2024          2109  3      end
;  2025          2110  2  else
;  2026          2111  3      begin
;  2027          2112  3      $release (rab = uafrab);
;  2028          2113  3      return false
;  2029          2114  3      end
```

```
; 2030          2115  1 end;


                                                                .EXTRN  SYS$UPDATE, SYS$MOD_IDENT
                                                                .EXTRN  SYS$RELEASE

                                                    07FC 00000 MODIFY_REC:
                                                                .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10                ; 1945
                                5A 00000000G    00   9E 00002          MOVAB   LIB$SIGNAL, R10                         ;
                                59 00000000'    00   9E 00009          MOVAB   DEFUSER+1, R9                           ;
                                58 00000000'    00   9E 00010          MOVAB   RECBUF+4, R8                            ;
                                5E        88    AE   9E 00017          MOVAB   -120(SP), SP                            ;
                          30    AE 010E0000     8F   D0 0001B          MOVL    #17694720, OLDACCDESC                   ; 1986
                                          34    AE   D4 00023          CLRL    OLDACCDESC+4                            ;
                          28    AE 010E0000     8F   D0 00026          MOVL    #17694720, NEWACCDESC                   ; 1987
                                          2C    AE   D4 0002E          CLRL    NEWACCDESC+4                            ;
                                1F      06CC    C8   E9 00031          BLBC    STR_WILD, 1$                            ; 2009
                                55        8C    A8   9E 00036          MOVAB   MATCH_TOKEN, R5                         ; 2011
                                53              68   9E 0003A          MOVAB   RECBUF+4, R3                            ;
                     68                 20      20   3A 0003D          LOCC    #32, #32, RECBUF+4                      ; 2010
                                52        E0    A0   9E 00041          MOVAB   -32(R0), R2                             ;
                                52              52   CE 00045          MNEGL   R2, R2                                  ;
                                54        D0    A8   D0 00048          MOVL    MATCH_TOKENLEN, R4                      ; 2011
                                   00000000G    00   16 0004C          JSB     FMG$MATCH_NAME                          ;
                                37              50   E9 00052          BLBC    R0, 3$                                  ;
                          06BC  C8              01   D0 00055 1$:      MOVL    #1, FOUND_MATCH                         ; 2014
                                54        FF    A9   9A 0005A          MOVZBL  DEFUSER, R4                             ; 2016
                                50        E0    A8   D0 0005E          MOVL    TOKENPTR, R0                            ;
     DC   A8                 20      69            54   2D 00062          CMPC5   R4, DEFUSER+1, #32, TOKENLEN, (R0)      ;
                                                    60      00068                                                       ;
                                                    10   13 00069          BEQL    2$                                  ;
                     68                 20          20   3A 0006B          LOCC    #32, #32, RECBUF+4                      ; 2018
                     50                 20          50   C3 0006F          SUBL3   R0, #32, R0                             ;
          50         20                 69          54   2D 00073          CMPC5   R4, DEFUSER+1, #32, R0, RECBUF+4        ; 2019
                                                    68      00078                                                       ;
                                                    14   12 00079          BNEQ    4$                                  ;
                          00000000'     00          01   D0 0007B 2$:      MOVL    #1, MOD_DEFAULT                         ; 2022
                          00000000V     00          00   FB 00082          CALLS   #0, DEFAULT_UAF                         ; 2023
                                              D8    A8   D6 00089          INCL    CALL_COUNT                              ; 2024
                                            0122    31   0008C 3$:       BRW     10$                                     ; 2025
          56                 02          1E          00   F0 0008F 4$:      INSV    #0, #30, #2, OLD_UIC                    ; 2031
          56                 0E          10      22  A8   F0 00094          INSV    RECBUF+38, #16, #14, OLD_UIC            ; 2032
                                          56      20  A8   B0 0009A          MOVW    RECBUF+36, OLD_UIC                      ; 2033
               58   AE          30      A8          20   28 0009E          MOVC3   #32, RECBUF+52, OLDACCNAME              ; 2034
               30   A8                  20          20   3A 000A4          LOCC    #32, #32, RECBUF+52                     ; 2035
               30   AE                  20          50   A3 000A9          SUBW3   R0, #32, OLDACCDESC                     ;
                              34    AE      58  AE   9E 000AE          MOVAB   OLDACCNAME, OLDACCDESC+4                ; 2036
                          00000000G     00          00   FB 000B3          CALLS   #0, UPDATE_RECORD                       ; 2038
                                        03          50   E8 000BA          BLBS    R0, 5$                                  ;
                                                  00F5   31 000BD          BRW     11$                                     ;
          57                 02          1E          00   F0 000C0 5$:      INSV    #0, #30, #2, NEW_UIC                    ; 2044
          57                 0E          10      22  A8   F0 000C5          INSV    RECBUF+38, #16, #14, NEW_UIC            ; 2045
                                          57      20  A8   B0 000CB          MOVW    RECBUF+36, NEW_UIC                      ; 2046
               38   AE          30      A8          20   28 000CF          MOVC3   #32, RECBUF+52, NEWACCNAME              ; 2047
               30   A8                  20          20   3A 000D5          LOCC    #32, #32, RECBUF+52                     ; 2048
               28   AE                  20          50   A3 000DA          SUBW3   R0, #32, NEWACCDESC                     ;
```

```
                          2C    AE        38    AE  9E 000DF          MOVAB   NEWACCNAME, NEWACCDESC+4                    ; 2049
                                    05E4    C8  9F 000E4          PUSHAB  UAFRAB                                      ; 2054
                00000000G   00            01  FB 000E8          CALLS   #1, SYS$UPDATE                              ;
                          F4    A8          50  D0 000EF          MOVL    R0, RMSERR                                  ;
                                13          50  E8 000F3          BLBS    R0, 6$                                      ;
                                      F4    A8  DD 000F6          PUSHL   RMSERR                                      ; 2057
                                            7E  D4 000F9          CLRL    -(SP)                                       ;
                                00000000G   8F  DD 000FB          PUSHL   #UAF$_MDFYERR                               ;
                                6A          03  FB 00101          CALLS   #3, LIB$SIGNAL                              ;
                                50        F4    A8  D0 00104          MOVL    RMSERR, R0                                  ; 2058
                                            04  00108          RET                                                 ;
                00000000'   00            01  D0 00109 6$:       MOVL    #1, MODIFY_FLAG                             ; 2062
                                00030002    8F  DD 00110          PUSHL   #196610                                     ; 2063
                00000000V   00            01  FB 00116          CALLS   #1, SECURITY_AUDIT                          ;
                                      D8    A8  D6 0011D          INCL    CALL_COUNT                                  ; 2064
                                    FF63    C9  9F 00120          PUSHAB  SD_MODIFY_IDENTIFIER                        ; 2067
                00000000G   00            01  FB 00124          CALLS   #1, CLI$PRESENT                             ;
                                70          50  E9 0012B          BLBC    R0, 8$                                      ;
                                7F        F0    A8  E9 0012E          BLBC    RDB_EXISTS, 10$                             ; 2068
                                57          56  D1 00132          CMPL    OLD_UIC, NEW_UIC                            ; 2075
                                            7A  13 00135          BEQL    10$                                         ;
                                6E  010E0000    8F  D0 00137          MOVL    #17694720, OLDIDDESC                        ; 2080
                                        04    AE  D4 0013E          CLRL    OLDIDDESC+4                                 ;
                                6E          20  B0 00141          MOVW    #32, OLDIDDESC                              ; 2082
                          04    AE        08    AE  9E 00144          MOVAB   OLDIDNAME, OLDIDDESC+4                      ; 2083
                                            7E  7C 00149          CLRQ    -(SP)                                       ; 2086
                                            7E  D4 0014B          CLRL    -(SP)                                       ;
                                        0C    AE  9F 0014D          PUSHAB  OLDIDDESC                                   ;
                                        10    AE  9F 00150          PUSHAB  OLDIDDESC                                   ;
                                            56  DD 00153          PUSHL   OLD_UIC                                     ;
                00000000G   00            06  FB 00155          CALLS   #6, SYS$IDTOASC                             ;
                                52          50  D0 0015C          MOVL    R0, STATUS                                  ;
                                17          52  E8 0015F          BLBS    STATUS, 7$                                  ; 2088
                                            52  DD 00162          PUSHL   STATUS                                      ; 2091
                                7E          56  3C 00164          MOVZWL  OLD_UIC, -(SP)                              ;
          7E          56                0E      10  EF 00167          EXTZV   #16, #14, OLD_UIC, -(SP)                    ; 2090
                                            02  DD 0016C          PUSHL   #2                                          ; 2089
                                00000000G   8F  DD 0016E          PUSHL   #UAF$_RDBMDFYERRU                           ;
                                6A          05  FB 00174          CALLS   #5, LIB$SIGNAL                              ;
                                            38  11 00177          BRB     10$                                         ;
                                            57  DD 00179 7$:       PUSHL   NEW_UIC                                     ; 2096
                                            7E  7C 0017B          CLRQ    -(SP)                                       ;
                                            7E  D4 0017D          CLRL    -(SP)                                       ;
                                            56  DD 0017F          PUSHL   OLD_UIC                                     ;
                    00000000G   00            05  FB 00181          CALLS   #5, SYS$MOD_IDENT                           ;
                                52          50  D0 00188          MOVL    R0, STATUS                                  ;
                                12          52  E8 0018B          BLBS    STATUS, 9$                                  ; 2098
                                            52  DD 0018E          PUSHL   STATUS                                      ; 2099
                                        04    AE  9F 00190          PUSHAB  OLDIDDESC                                   ;
                                            01  DD 00193          PUSHL   #1                                          ;
                                00000000G   8F  DD 00195          PUSHL   #UAF$_RDBMDFYERR                            ;
                                6A          04  FB 00198          CALLS   #4, LIB$SIGNAL                              ;
                                            11  11 0019E 8$:       BRB     10$                                         ;
                          F8    A8            01  90 001A0 9$:       MOVB    #1, RIGHTSLIST_MODIFIED                     ; 2102
                                            5E  DD 001A4          PUSHL   SP                                          ; 2103
                                            01  DD 001A6          PUSHL   #1                                          ;
                                00000000G   8F  DD 001A8          PUSHL   #UAF$_RDBMDFYMSG                            ;
```

```
                                 6A           03  FB 001AE         CALLS   #3, LIB$SIGNAL                                    ;
                                 50           01  D0 001B1 10$:    MOVL    #1, R0                                           ; 2111
                                                  04 001B4         RET                                                      ;
                                                                                                                           ;
                                      05E4    C8  9F 001B5 11$:    PUSHAB  UAFRAP                                           ; 2112
                          00000000G  00           01  FB 001B9    CALLS   #1, SYS$RELEASE                                  ;
                                                  50  D4 001C0    CLRL    R0                                               ; 2113
                                                      04 001C2    RET                                                      ; 2115

; Routine Size:  451 bytes,    Routine Base:  $CODE$ + 0990
```

```
;2032     2116  1 %sbttl 'remove_uaf - remove username from file'
;2033     2117  1 global routine remove_uaf : novalue =
;2034     2118  2 begin
;2035     2119  2
;2036     2120  2 !++
;2037     2121  2 !
;2038     2122  2 ! FUNCTIONAL DESCRIPTION:
;2039     2123  2 !
;2040     2124  2 !       Routine to delete a user record from the UAF file.
;2041     2125  2 !
;2042     2126  2 ! INPUTS:
;2043     2127  2 !
;2044     2128  2 !       none
;2045     2129  2 !
;2046     2130  2 ! OUTPUTS:
;2047     2131  2 !
;2048     2132  2 !       none
;2049     2133  2 !
;2050     2134  2 ! IMPLICIT INPUTS:
;2051     2135  2 !
;2052     2136  2 !       rename: a logical flag which indicates whether this COPY is part
;2053     2137  2 !               of a RENAME operation
;2054     2138  2 !
;2055     2139  2 ! ROUTINE VALUE:
;2056     2140  2 !
;2057     2141  2 !       none
;2058     2142  2 !
;2059     2143  2 ! SIDE EFFECTS:
;2060     2144  2 !
;2061     2145  2 !  This routine also deletes any NETUAF entries for the
;2062     2146  2 !       local user which is to be removed (If NETUAF.DAT exists)
;2063     2147  2 !
;2064     2148  2 !--
;2065     2149  2
;2066     2150  2 !
;2067     2151  2 ! Look for the record specified by the user.  Make sure
;2068     2152  2 ! the DEFAULT and SYSTEM records are not removed.
;2069     2153  2 !
;2070     2154  2
;2071     2155  2 if get_user_record (true, false)
;2072     2156  2 then
;2073     2157  2
;2074     2158  2 !
;2075     2159  2 ! User record has been found and read into RECBUF
;2076     2160  2 ! Zero the username field and update the record in the file
;2077     2161  2 !
;2078     2162  3     begin
;2079     2163  4     if rmsbad ($delete (rab=uafrab))
;2080     2164  3     then
;2081     2165  3         LIB$SIGNAL(UAF$_REMERR, 0, .rmserr)
;2082     2166  3     e  e
;2083     2167  4         begin
;2084     2168  4         modify_flag = true;                        ! mark file as modified
;2085     2169  4         if not .uaf$gl_ctlmsk[uaf$v_rename]
;2086     2170  4         then
;2087     2171  4             security_audit (nsa$k_recid_sysuaf_del);
;2088     2172  4         if not .uaf$gl_ctlmsk[uaf$v_rename]
```

```
;  2089          2173  4          then
;  2090          2174  5              begin
;  2091          2175  5              !
;  2092          2176  5              ! If there is a proxy login file, delete entries for
;  2093          2177  5              ! the user just removed from SYSUAF.DAT
;  2094          2178  5              !
;  2095          2179  5              if .netuaf_exists
;  2096          2180  5              then adjust_proxy (remove_records);
;  2097          2181  5              LIB$SIGNAL(UAF$_REMMSG);
;  2098          2182  5              !
;  2099          2183  5              ! Unless specifically requested not to
;  2100          2184  5              ! remove the corresponding identifier from the
;  2101          2185  5              ! rights data base.
;  2102          2186  5              !
;  2103          2187  6              if (cli$present ( sd_remove_identifier ) and
;  2104          2188  6                  .rdb_exists )
;  2105          2189  5              then
;  2106          2190  5                  uaf$remove_ident_recbuf () ;
;  2107          2191  4              end;
;  2108          2192  3          end;
;  2109          2193  2      end;
;  2110          2194  1  end;


                                                        .EXTRN  SYS$DELETE

                                        000C 00000      .ENTRY  REMOVE_UAF, Save R2,R3                          ; 2117
                    53 00000000G   00    9E 00002      MOVAB   LIB$SIGNAL, R3                                  ;
                    52 00000000'   00    9E 00009      MOVAB   RMSERR, R2                                      ;
                               7E   01    7D 00010      MOVQ    #1, -(SP)                                       ; 2155
          00000000V            00   02    FB 00013      CALLS   #2, GET_USER_RECORD                             ;
                               6C   50    E9 0001A      BLBC    R0, 3$                                          ;
                         05F0       C2    9F 0001D      PUSHAB  UAFRAB                                          ; 2163
          00000000G            00   01    FB 00021      CALLS   #1, SYS$DELETE                                  ;
                               62   50    D0 00028      MOVL    R0, RMSERR                                      ;
                               0E   50    E8 0002B      BLBS    R0, 1$                                          ;
                                    62    DD 0002E      PUSHL   RMSERR                                          ; 2165
                                    7E    D4 00030      CLRL    -(SP)                                           ;
                         00000000G  8F    DD 00032      PUSHL   #UAF$_REMERR                                    ;
                               63   03    FB 00038      CALLS   #3, LIB$SIGNAL                                  ;
                                    04    0003B      RET                                                     ;
          00000000'            00   01    D0 0003C 1$:  MOVL    #1, MODIFY_FLAG                                 ; 2168
                               42       8C A2    E8 00043      BLBS    UAF$GL_CTLMSK, 3$                       ; 2169
                         00020002   8F    DD 00047      PUSHL   #131074                                         ; 2171
          00000000V            00   01    FB 0004D      CALLS   #1, SECURITY_AUDIT                              ;
                               31       8C A2    E8 00054      BLBS    UAF$GL_CTLMSK, 3$                       ; 2172
                               09       F8 A2    E9 00058      BLBC    NETUAF_EXISTS, 2$                       ; 2179
                                    01    DD 0005C      PUSHL   #1                                              ; 2180
          00000000V            00   01    FB 0005E      CALLS   #1, ADJUST_PROXY                                ;
                         00000000G  8F    DD 00065 2$:  PUSHL   #UAF$_REMMSG                                    ; 2181
                               63   01    FB 0006B      CALLS   #1, LIB$SIGNAL                                  ;
                         00000000'  00    9F 0006E      PUSHAB  SD_REMOVE_IDENTIFIER                            ; 2187
          00000000G            00   01    FB 00074      CALLS   #1, CLI$PRESENT                                 ;
                               0B   50    E9 0007B      BLBC    R0, 3$                                          ;
                               07       FC A2    E9 0007E      BLBC    RDB_EXISTS, 3$                          ; 2188
          00000000G            00   00    FB 00082      CALLS   #0, UAF$REMOVE_IDENT_RECBUF                     ; 2190
```

```
                                        04 00089 3$:     RET                                                    ; 2194

; Routine Size:  138 bytes,    Routine Base:  $CODE$ + 0B53
```

```
;  2112          2195  1 %sbttl 'remove_proxy - remove a proxy record'
;  2113          2196  1 global routine remove_proxy : novalue =
;  2114          2197  2 begin
;  2115          2198  2
;  2116          2199  2 !++
;  2117          2200  2 !
;  2118          2201  2 ! FUNCTIONAL CHARACTERISTICS:
;  2119          2202  2 !
;  2120          2203  2 !       This routine removes proxy login entries from NETUAF.DAT
;  2121          2204  2 !
;  2122          2205  2 ! INPUTS:
;  2123          2206  2 !
;  2124          2207  2 !       none
;  2125          2208  2 !
;  2126          2209  2 ! OUTPUTS:
;  2127          2210  2 !
;  2128          2211  2 !       none
;  2129          2212  2 !
;  2130          2213  2 ! IMPLICIT INPUTS:
;  2131          2214  2 !
;  2132          2215  2 !       none
;  2133          2216  2 !
;  2134          2217  2 ! IMPLICIT OUTPUTS:
;  2135          2218  2 !
;  2136          2219  2 !       none
;  2137          2220  2 !
;  2138          2221  2 ! ROUTINE VALUE:
;  2139          2222  2 !
;  2140          2223  2 !       none
;  2141          2224  2 !
;  2142          2225  2 ! SIDE EFFECTS:
;  2143          2226  2 !
;  2144          2227  2 !       An entry is removed from NETUAF.DAT
;  2145          2228  2 !
;  2146          2229  2 !--
;  2147          2230  2
;  2148          2231  2 local
;  2149          2232  2         node_len,
;  2150          2233  2         node_ptr,
;  2151          2234  2         remuser_len,
;  2152          2235  2         remuser_ptr,
;  2153          2236  2         counter,
;  2154          2237  2         success;
;  2155          2238  2
;  2156          2239  2 !
;  2157          2240  2 ! Make sure NETUAF.DAT exists
;  2158          2241  2 !
;  2159          2242  2 if not .netuaf_exists
;  2160          2243  2 then return LIB$SIGNAL(UAF$_NAFDNE);
;  2161          2244  2
;  2162          2245  2 !
;  2163          2246  2 ! Retrieve remote name in node::remotename form
;  2164          2247  2 !
;  2165          2248  2 cli$get_value (sd_token1, tokendsc);
;  2166          2249  2
;  2167          2250  2 !
;  2168          2251  2 ! Verify proper format
```

```
;  2169          2252  2 !
;  2170          2253  2 if not remote_parse (node_ptr, node_len, remuser_ptr, remuser_len)
;  2171          2254  2 then return;
;  2172          2255  2
;  2173          2256  2 !
;  2174          2257  2 ! Copy into appropriate fields
;  2175          2258  2 !
;  2176          2259  2 ch$copy (.node_len, .node_ptr, ' ', naf$s_node, netbuf[naf$t_node]);
;  2177          2260  2 ch$copy (.remuser_len, .remuser_ptr, ' ', naf$s_remuser, netbuf[naf$t_remuser]);
;  2178          2261  2
;  2179          2262  2 nafrab[rab$l_rop] = rab$m_rlk;
;  2180          2263  2
;  2181          2264  2 success = get_proxy_record ();
;  2182          2265  2
;  2183          2266  2 !
;  2184          2267  2 ! Delete the record
;  2185          2268  2 !
;  2186          2269  2 if .success
;  2187          2270  2 then
;  2188          2271  3     begin
;  2189          2272  4     if rmsbad ($delete (rab = nafrab))
;  2190          2273  3     then
;  2191          2274  3         LIB$SIGNAL(UAF$_REMERR, 0, .rmserr)
;  2192          2275  3     else
;  2193          2276  4         begin
;  2194          2277  4         security_audit (nsa$k_recid_netuaf_del);
;  2195          2278  4         LIB$SIGNAL(UAF$_PREMMSG);
;  2196          2279  3         end;
;  2197          2280  3     end
;  2198          2281  2 else
;  2199          2282  2     LIB$SIGNAL(UAF$_REMERR);
;  2200          2283  2
;  2201          2284  2 netuaf_modified = true;
;  2202          2285  1 end;


                                    01FC 00000          .ENTRY  REMOVE_PROXY, Save R2,R3,R4,R5,R6,R7,R8     ; 2196
                          58 00000000G  8F   DO 00002   MOVL    #UAF$_REMERR, R8                            ;
                          57 00000000G  00   9E 00009   MOVAB   LIB$SIGNAL, R7                              ;
                          56 00000000'  00   9E 00010   MOVAB   RMSERR, R6                                  ;
                          5E            10   C2 00017   SUBL2   #16, SP                                    ;
                          0A      F8    A6   E8 0001A   BLBS    NETUAF_EXISTS, 1$                           ; 2242
                             00000000G  8F   DD 0001E   PUSHL   #UAF$_NAFDNE                                ; 2243
                          67            01   FB 00024   CALLS   #1, LIB$SIGNAL                            ;
                                        04   00027   RET                                                  ;
                                E8      A6   9F 00028 1$: PUSHAB  TOKENDSC                                ; 2248
                             00000000'  00   9F 0002B   PUSHAB  SD_TOKEN1                                  ;
                    00000000G  00               02   FB 00031   CALLS   #2, CLI$GET_VALUE                  ;
                                                5E   DD 00038   PUSHL   SP                                  ; 2253
                                          08    AE   9F 0003A   PUSHAB  REMUSER_PTR                         ;
                                          10    AE   9F 0003D   PUSHAB  NODE_LEN                            ;
                                          18    AE   9F 00040   PUSHAB  NODE_PTR                            ;
                                FA17      CF        04   FB 00043   CALLS   #4, REMOTE_PARSE                ;
                                          63        50   E9 00048   BLBC    R0, 6$                          ;
```

```
                20              20      0C      BE      08      AE  2C 0004B           MOVC5   NODE_LEN, @NODE_PTR, #32, #32, NETBUF       ; 2259
                                                        058C    C6     00052
                20              20      04      BE              6E  2C 00055           MOVC5   REMUSER_LEN, @REMUSER_PTR, #32, #32, -       ; 2260
                                                        C5AC    C6     0005B                   NETBUF+32
                                    0638    C6  00080000        8F  D0 0005E           MOVL    #524288, NAFRAB+4                           ; 2262
                            00000000V   00                      00  FB 00067           CALLS   #0, GET_PROXY_RECORD                        ; 2264
                                        31                      50  E9 0006E           BLBC    SUCCESS, 3$                                 ; 2269
                                                        0634    C6  9F 00071           PUSHAB  NAFRAB                                      ; 2272
                            00000000G   00                      01  FB 00075           CALLS   #1, SYS$DELETE
                                        66                      50  D0 0007C           MOVL    R0, RMSERR
                                        0B                      50  E8 0007F           BLBS    R0, 2$
                                                                66  DD 00082           PUSHL   RMSERR                                      ; 2274
                                                                7E  D4 00084           CLRL    -(SP)
                                                                58  DD 00086           PUSHL   R8
                                        67                      03  FB 00088           CALLS   #3, LIB$SIGNAL
                                                                1A  11 0008B           BRB     5$
                                            00020003            8F  DD 0008D 2$:       PUSHL   #131075                                     ; 2277
                            00000000V   00                      01  FB 00093           CALLS   #1, SECURITY_AUDIT
                                            00000000G           8F  DD 0009A           PUSHL   #UAF$_PREMMSG                               ; 2278
                                                                02  11 000A0           BRB     4$
                                                                58  DD 000A2 3$:       PUSHL   R8                                          ; 2282
                                        67                      01  FB 000A4 4$:       CALLS   #1, LIB$SIGNAL
                            00000000'   00                      01  D0 000A7 5$:       MOVL    #1, NETUAF_MODIFIED                         ; 2284
                                                                    04 000AE 6$:       RET                                                 ; 2285

; Routine Size:  175 bytes,    Routine Base:  $CODE$ + 0BDD
```

```
;  2204          2286  1 %sbttl 'rename_uaf - rename user record'
;  2205          2287  1 global routine rename_uaf : novalue =
;  2206          2288  2 begin
;  2207          2289  2
;  2208          2290  2 !++
;  2209          2291  2 !
;  2210          2292  2 ! FUNCTIONAL DESCRIPTION:
;  2211          2293  2 !
;  2212          2294  2 !       Effect a user authorization record rename, by
;  2213          2295  2 !       performing a COPY and a REMOVE operation.
;  2214          2296  2 !
;  2215          2297  2 ! INPUTS:
;  2216          2298  2 !
;  2217          2299  2 !       none
;  2218          2300  2 !
;  2219          2301  2 ! IMPLICIT INPUTS:
;  2220          2302  2 !
;  2221          2303  2 !       none
;  2222          2304  2 !
;  2223          2305  2 ! OUTPUTS:
;  2224          2306  2 !
;  2225          2307  2 !       none
;  2226          2308  2 !
;  2227          2309  2 ! IMPLICIT OUTPUTS:
;  2228          2310  2 !
;  2229          2311  2 !       none
;  2230          2312  2 !
;  2231          2313  2 ! SIDE EFFECTS:
;  2232          2314  2 !
;  2233          2315  2 !       A user authorization record is copied with a newname; the
;  2234          2316  2 !       original record is then deleted.
;  2235          2317  2 !       This routine also causes updating of any NETUAF entries for the
;  2236          2318  2 !       local user which is to be renamed.
;  2237          2319  2 !
;  2238          2320  2 !--
;  2239          2321  2
;  2240          2322  2 uaf$gl_ctlmsk[uaf$v_rename] = true;
;  2241          2323  2 !
;  2242          2324  2 ! Copy the new authorization record
;  2243          2325  2 !
;  2244          2326  3 if  (copy_uaf ())
;  2245          2327  2 then
;  2246          2328  3     begin
;  2247          2329  3 !
;  2248          2330  3 ! Remove the old authorization record
;  2249          2331  3 !
;  2250          2332  3     remove_uaf ();
;  2251          2333  3     LIB$SIGNAL(UAF$_RENMSG);
;  2252          2334  3 !
;  2253          2335  3 ! Because passwords are folded in with the username, passwords for
;  2254          2336  3 ! RENAMEd records will no longer work--warn the user
;  2255          2337  3 !
;  2256          2338  3     if .pwd_flag
;  2257          2339  3     then LIB$SIGNAL(UAF$_DEFPWD);
;  2258          2340  3
;  2259          2341  3 !
;  2260          2342  3 ! Modify the rights data base if one exists
```

```
;  2261          2343  3 !
;  2262          2344  4     if (cli$present ( sd_modify_identifier ) and
;  2263          2345  4         .rdb_exists )
;  2264          2346  3     then
;  2265          2347  4         begin
;  2266          2348  4         local
;  2267          2349  4             status              : long,
;  2268          2350  4             old_name_buff       : vector [kgb$s_name, byte],
;  2269          2351  4             old_name_desc       : statdesc ,
;  2270          2352  4             new_name_desc       : statdesc ;
;  2271          2353  4
;  2272          2354  4         old_name_desc[length]  = kgb$s_name ;
;  2273          2355  4         old_name_desc[pointer] = old_name_buff ;
;  2274          2356  4         new_name_desc[length]  = .newuser[en ;
;  2275          2357  4         new_name_desc[pointer] = newusername ;
;  2276          2358  4
;  2277          2359  4         uaf$find_uic () ;       !  Build Identifier from the recbuf
;  2278          2360  4
;  2279          2361  4         !
;  2280          2362  4         ! Find the ascii name
;  2281          2363  4         !
;  2282        P 2364  4         status = $idtoasc ( id     = .ident,
;  2283        P 2365  4                             namlen = old_name_desc[length],
;  2284          2366  4                             nambuf = old_name_desc ) ;
;  2285          2367  4         if not .status
;  2286          2368  4         then LIB$SIGNAL(UAF$_RDBMDFYERRU, 2,
;  2287          2369  4                            .ident[uic$v_group],
;  2288          2370  4                            .ident[uic$v_member], .status)
;  2289          2371  4         else
;  2290          2372  5             begin
;  2291          2373  5
;  2292        P 2374  5             status = $mod_ident ( id       = .ident,
;  2293          2375  5                                   new_name = new_name_desc ) ;
;  2294          2376  5
;  2295          2377  5             if not .status
;  2296          2378  5                 then LIB$SIGNAL(UAF$_RDBMDFYERR, 1, old_name_desc, .status)
;  2297          2379  5                 else
;  2298          2380  6                     begin
;  2299          2381  6                     rightslist_modified = true ;
;  2300          2382  6                     LIB$SIGNAL(UAF$_RDBMDFYMSG, 1, old_name_desc);
;  2301          2383  5                     end ;
;  2302          2384  4             end ;
;  2303          2385  3         end ;
;  2304          2386  3
;  2305          2387  2     end;
;  2306          2388  2 !
;  2307          2389  2 ! Reset flags
;  2308          2390  2 !
;  2309          2391  2 rename_ph2 = false;
;  2310          2392  2 uaf$gl_ctlmsk[uaf$v_rename] = false;
;  2311          2393  2
;  2312          2394  1 end;
```

```
                                 007C 00000          .ENTRY  RENAME_UAF, Save R2,R3,R4,R5,R6                 ; 2287
                          56 00000000'   00   9E 00002       MOVAB   NEWUSERLEN, R6
                          55 00000000G   00   9E 00009       MOVAB   IDENT, R5
                          54 00000000'   00   9E 00010       MOVAB   UAF$GL_CTLMSK, R4
                          53 00000000G   00   9E 00017       MOVAB   LIB$SIGNAL, R3
                          5E             30   C2 0001E       SUBL2   #48, SP
                          64             01   88 00021       BISB2   #1, UAF$GL_CTLMSK                       ; 2322
                    FA08  CF             00   FB 00024       CALLS   #0, COPY_UAF                            ; 2326
                          29             50   E9 00029       BLBC    R0, 2$
                    FE96  CF             00   FB 0002C       CALLS   #0, REMOVE_UAF                          ; 2332
                          00000000G      8F   DD 00031       PUSHL   #UAF$_RENMSG                            ; 2333
                          63             01   FB 00037       CALLS   #1, LIB$SIGNAL
                          09     06F4    C4   E9 0003A       BLBC    PWD_FLAG, 1$                            ; 2338
                          00000C000G     8F   DD 0003F       PUSHL   #UAF$_DEFPWD                            ; 2339
                          63             01   FB 00045       CALLS   #1, LIB$SIGNAL
                          00000000'      00   9F 00048 1$:   PUSHAB  SD_MODIFY_IDENTIFIER                    ; 2344
                00000000G 00             01   FB 0004E       CALLS   #1, CLI$PRESENT
                          60             50   E9 00055 2$:   BLBC    R0, 3$
                          5C       70    A4   E9 00058       BLBC    RDB_EXISTS, 3$                          ; 2345
                       08 AE 010E0000    8F   D0 0005C       MOVL    #17694720, OLD_NAME_DESC                ; 2351
                          0C             AE   D4 00064       CLRL    OLD_NAME_DESC+4
                          6E 010E0000    8F   D0 00067       MOVL    #17694720, NEW_NAME_DESC                ; 2352
                             04          AE   D4 0006E       CLRL    NEW_NAME_DESC+4
                       08 AE             20   B0 00071       MOVW    #32, OLD_NAME_DESC                      ; 2354
                       0C AE       10    AE   9E 00075       MOVAB   OLD_NAME_BUFF, OLD_NAME_DESC+4          ; 2355
                          6E             66   B0 0007A       MOVW    NEWUSERLEN, NEW_NAME_DESC               ; 2356
                       04 AE       04    A6   9E 0007D       MOVAB   NEWUSERNAME, NEW_NAME_DESC+4            ; 2357
                00000000G 00             00   FB 00082       CALLS   #0, UAF$FIND_UIC                        ; 2359
                                         7E   7C 00089       CLRQ    -(SP)                                   ; 2366
                                         7E   D4 0008B       CLRL    -(SP)
                             14          AE   9F 0008D       PUSHAB  OLD_NAME_DESC
                             18          AE   9F 00090       PUSHAB  OLD_NAME_DESC
                                         65   DD 00093       PUSHL   IDENT
                      00000000G  00            06   FB 00095       CALLS   #6, SYS$IDTOASC
                                 52            50   D0 0009C       MOVL    R0, STATUS
                                 18            52   E8 0009F       BLBS    STATUS, 4$                          ; 2367
                                               52   DD 000A2       PUSHL   STATUS                          ; 2370
                                 7E            65   3C 000A4       MOVZWL  IDENT, -(SP)
                7E       02  A5                0E   00   EF 000A7       EXTZV   #0, #14, IDENT+2, -(SP)         ; 2369
                                               02   DD 000AD       PUSHL   #2                              ; 2368
                                      00000000G  8F   DD 000AF       PUSHL   #UAF$_RDBMDFYERRU
                                 63            05   FB 000B5       CALLS   #5, LIB$SIGNAL
                                               3A   11 000B8 3$:   BRB     6$
                                               7E   D4 000BA 4$:   CLRL    -(SP)                           ; 2375
                                                   04   AE   9F 000BC       PUSHAB  NEW_NAME_DESC
                                               7E   7C 000BF       CLRQ    -(SP)
                                               65   DD 000C1       PUSHL   IDENT
                      00000000G  00            05   FB 000C3       CALLS   #5, SYS$MOD_IDENT
                                 52            50   D0 000CA       MOVL    R0, STATUS
                                 12            52   E8 000CD       BLBS    STATUS, 5$                          ; 2377
                                               52   DD 000D0       PUSHL   STATUS                          ; 2378
                                                   0C   AE   9F 000D2       PUSHAB  OLD_NAME_DESC
                                               01   DD 000D5       PUSHL   #1
                                      00000000G  8F   DD 000D7       PUSHL   #UAF$_RDBMDFYERR
                                 63            04   FB 000DD       CALLS   #4, LIB$SIGNAL
                                               12   11 000E0       BRB     6$
                             78  A4                01   90 000E2 5$:   MOVB    #1, RIGHTSLIST_MODIFIED         ; 2381
```

```
                            08    AE  9F 000E6          PUSHAB  OLD_NAME_DESC                          ; 2382
                                  01  DD 000E9          PUSHL   #1                                    ;
                      00000000G   8F  DD 000EB          PUSHL   #UAF$_RDBMDFYMSG                      ;
                63                03  FB 000F1          CALLS   #3, LIB$SIGNAL                        ;
                            D8    A6  94 000F4 6$:       CLRB    RENAME_PH2                            ; 2391
                64                01  8A 000F7          BICB2   #1, UAF$GL_CTLMSK                     ; 2392
                                      04 000FA          RET                                           ; 2394

; Routine Size:  251 bytes,    Routine Base:  $CODE$ + 0C8C
```

```
;2314    2395  1 %sbttl 'adjust_proxy - implicitly remove/update proxy record'
;2315    2396  1 global routine adjust_proxy (remove) : novalue =
;2316    2397  2 begin
;2317    2398  2
;2318    2399  2 !++
;2319    2400  2 !
;2320    2401  2 ! FUNCTIONAL DESCRIPTION:
;2321    2402  2 !
;2322    2403  2 !       This routine performs the operations implicitly indicated by
;2323    2404  2 !       REMOVE or RENAME operations on SYSUAF.DAT.  If a SYSUAF.DAT
;2324    2405  2 !       record is removed, then any corresponding NETUAF.DAT entries must
;2325    2406  2 !       also be deleted.  If a SYSUAF.DAT record is renamed, then any
;2326    2407  2 !       corresponding NETUAF.DAT entries must be updated.
;2327    2408  2 !
;2328    2409  2 ! INPUTS:
;2329    2410  2 !
;2330    2411  2 !       remove - a flag which indicates that the NETUAF.DAT record
;2331    2412  2 !                should be removed.  If false, then the record
;2332    2413  2 !                should be updated.
;2333    2414  2 !
;2334    2415  2 ! OUTPUTS:
;2335    2416  2 !
;2336    2417  2 !       none
;2337    2418  2 !
;2338    2419  2 ! IMPLICIT INPUTS:
;2339    2420  2 !
;2340    2421  2 !       olduserlen  - the old username length for a RENAME
;2341    2422  2 !       oldusername - the old username string for a RENAME
;2342    2423  2 !       TOKENPTR - the new username for RENAME, the removed username for REMOVE
;2343    2424  2 !       TOKENLEN - "    "     "      length  "  ,    "      "        "   length  "  "
;2344    2425  2 !
;2345    2426  2 ! IMPLICIT OUTPUTS:
;2346    2427  2 !
;2347    2428  2 !       none
;2348    2429  2 !
;2349    2430  2 ! SIDE EFFECTS:
;2350    2431  2 !
;2351    2432  2 !        A  record is removed/updated in NETUAF.DAT
;2352    2433  2 !
;2353    2434  2 !--
;2354    2435  2
;2355    2436  2 local
;2356    2437  2         modified: initial(false),
;2357    2438  2         status;
;2358    2439  2
;2359    2440  2 !
;2360    2441  2 ! Set access to sequential because we need to check for
;2361    2442  2 ! multiple entries
;2362    2443  2 !
;2363    2444  2 nafrab[rab$l_rop] = rab$m_rlk;          ! Lock records for writing
;2364    2445  2 nafrab[rab$b_rac] = rab$c_seq;
;2365    2446  2 $rewind (rab = nafrab);
;2366    2447  2
;2367    2448  2 !
;2368    2449  2 ! Until EOF...
;2369    2450  2 !
;2370    2451  2 while status = get_proxy_record ()
```

```
; 2371          2452  2 do
; 2372          2453  2     begin
; 2373          2454  2     local
; 2374          2455  2         locuser_len,
; 2375          2456  2         blank_ptr;
; 2376          2457  2
; 2377          2458  2 !
; 2378          2459  2 ! Find end of user name...
; 2379          2460  2 !
; 2380          2461  2 !
; 2381          2462  2 ! If not found in 12 characters, it must be the full 12
; 2382          2463  2 ! characters in length
; 2383          2464  2 !
; 2384          2465  2     if ch$fail (blank_ptr = ch$find_ch (naf$s_localuser,
; 2385          2466  2                                        netbuf[naf$t_localuser], ' '))
; 2386          2467  2     then
; 2387          2468  2         locuser_len = naf$s_localuser
; 2388          2469  2     else
; 2389          2470  2         locuser_len = .blank_ptr - netbuf[naf$t_localuser];
; 2390          2471  2
; 2391          2472  2 !
; 2392          2473  2 ! If this is a record to be removed, delete it
; 2393          2474  2 !
; 2394          2475  2     if .remove
; 2395          2476  2     then
; 2396          2477  4         begin
; 2397          2478  4         if .tokenlen eql .locuser_len
; 2398          2479  4         and  ch$eql (.tokenlen, .tokenptr, .locuser_len, netbuf[naf$t_localuser])
; 2399          2480  4         then
; 2400          2481  5             begin
; 2401          2482  5             netuaf_modified = true;
; 2402          2483  5             $delete (rab = nafrab);
; 2403          2484  5             security_audit (nsa$k_recid_netuaf_del);
; 2404          2485  4             end;
; 2405          2486  4         end
; 2406          2487  4 !
; 2407          2488  4 ! otherwise, change the localusername field to reflect the
; 2408          2489  4 ! new username in SYSUAF.DAT
; 2409          2490  4 !
; 2410          2491  3     else
; 2411          2492  4         begin
; 2412          2493  4         modified = true;
; 2413          2494  4         if .olduserlen eql .locuser_len
; 2414          2495  4         and  ch$eql (.olduserlen, oldusername,
; 2415          2496  4                      .locuser_len, netbuf[naf$t_localuser])
; 2416          2497  4         then
; 2417          2498  5             begin
; 2418          2499  5             ch$copy (uaf$s_username, recbuf[uaf$t_username],' ',
; 2419          2500  5                      naf$s_localuser,netbuf[naf$t_localuser]);
; 2420          2501  5             $update (rab = nafrab);
; 2421          2502  5             netuaf_modified = true;
; 2422          2503  5             security_audit (nsa$k_recid_netuaf_mod, oldusername);
; 2423          2504  4             end;
; 2424          2505  3         end;
; 2425          2506  2     end;
; 2426          2507  2
; 2427          2508  2 if .modified then
```

```
;  2428          2509  2     LIB$SIGNAL(UAF$_ZZPRACREN, 2, .olduserlen, oldusername);
;  2429          2510  2
;  2430          2511  2 !
;  2431          2512  2 ! Return to keyed access
;  2432          2513  2 !
;  2433          2514  2 nafrab[rab$b_rac] = rab$c_key;
;  2434          2515  1 end;


                                                                .EXTRN  SYS$REWIND

                                                 07FC 00000     .ENTRY  ADJUST_PROXY, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 2396
                                                                R10
                                5A 00000000V     00   9E 00002 MOVAB   SECURITY_AUDIT, R10
                                59 00000000'     00   9E 00009 MOVAB   OLDUSERNAME, R9
                                58 00000000'     00   9E 00010 MOVAB   NETBUF+64, R8
                                                 56   D4 00017 CLRL    MODIFIED                                  ; 2397
                          6C    A8 00080000      8F   D0 00019 MOVL    #524288, NAFRAB+4                         ; 2444
                                       0086      C8   94 00021 CLRB    NAFRAB+30                                 ; 2445
                                         68      A8   9F 00025 PUSHAB  NAFRAB                                    ; 2446
                    00000000G   00               01   FB 00028 CALLS   #1, SYS$REWIND
                    00000000V   00               00   FB 0002F 1$:    CALLS   #0, GET_PROXY_RECORD              ; 2451
                                57               50   D0 00036 MOVL    R0, STATUS
                                03               57   E8 00039 BLBS    STATUS, 2$
                                                 0081 31 0003C BRW     8$
                          68                     20   3A 0003F 2$:    LOCC    #32, #32, NETBUF+64               ; 2466
                                                 02   12 00043 BNEQ    3$
                                                 51   D4 00045 CLRL    R1
                                                 51   D5 00047 3$:    TSTL    BLANK_PTR
                                                 05   12 00049 BNEQ    4$
                                50               20   D0 0004B MOVL    #32, LOCUSER_LEN                          ; 2468
                                                 07   11 0004E BRB     5$
                                52               68   9E 00050 4$:    MOVAB   NETBUF+64, R2                     ; 2470
                          50    51               52   C3 00053 SUBL3   R2, BLANK_PTR, LOCUSER_LEN
                                30         04    AC   E9 00057 5$:    BLBC    REMOVE, 7$                        ; 2478
                                51       FA1C    C8   3C 0005B MOVZWL  TOKENLEN, R1
                                50               51   D1 00060 CMPL    R1, LOCUSER_LEN
                                                 CA   12 00063 BNEQ    1$
                                52       FA20    C8   D0 00065 MOVL    TOKENPTR, R2                              ; 2479
                    50    00    62               51   2D 0006A CMPC5   R1, (R2), #0, LOCUSER_LEN, NETBUF+64
                                                 68        0006F
                                                 BD   12 00070 BNEQ    1$
                                F4    A9         01   D0 00072 MOVL    #1, NETUAF_MODIFIED                       ; 2482
                                         68      A8   9F 00076 PUSHAB  NAFRAB                                    ; 2483
                          00000000G   00         01   FB 00079 CALLS   #1, SYS$DELETE
                                      00020003   8F   DD 00080 PUSHL   #131075                                    ; 2484
                                6A               01   FB 00086 CALLS   #1, SECURITY_AUDIT
                                                 A4   11 00089 6$:    BRB     1$                                ; 2475
                                56               01   D0 0008B 7$:    MOVL    #1, MODIFIED                      ; 2493
                                51         FC    A9   D0 0008E MOVL    OLDUSERLEN, R1                            ; 2494
                                50               51   D1 00092 CMPL    R1, LOCUSER_LEN
                                                 98   12 00095 BNEQ    1$
                    50    00    69               51   2D 00097 CMPC5   R1, OLDUSERNAME, #0, LOCUSER_LEN, NETBUF+64 ; 2496
                                                 68        0009C
                                                 90   12 0009D BNEQ    1$
                          68    FA40       C8          20   28 0009F MOVC3   #32, RECBUF+4, NETBUF+64             ; 2500
```

```
                                    68    A8  9F 000A5          PUSHAB  NAFRAB                                  ; 2501
                        00000000G   00        01  FB 000A8     CALLS   #1, SYS$UPDATE                         ;
                               F4   A9        01  D0 000AF     MOVL    #1, NETUAF_MODIFIED                    ; 2502
                                              59  DD 000B3     PUSHL   R9                                     ; 2503
                                    00030003  8F  DD 000B5     PUSHL   #196611                                ;
                                   6A         02  FB 000BB     CALLS   #2, SECURITY_AUDIT                     ;
                                              C9  11 000BE     BRB     6$                                     ; 2451
                                   14         56  E9 000C0 8$: BLBC    MODIFIED, 9$                           ; 2508
                                              59  DD 000C3     PUSHL   R9                                     ; 2509
                                        FC    A9  DD 000C5     PUSHL   OLDUSERLEN                             ;
                                              02  DD 000C8     PUSHL   #2                                     ;
                                    00000000G 8F  DD 000CA     PUSHL   #UAF$_ZZPRACREN                        ;
                        00000000G   00        04  FB 000D0     CALLS   #4, LIB$SIGNAL                         ;
                              0086  C8        01  90 000D7 9$: MOVB    #1, NAFRAB+30                          ; 2514
                                              04  000DC       RET                                            ; 2515

; Routine Size:  221 bytes,    Routine Base:  $CODE$ + 0D87
```

```
;  2436     2516  1 %sbttl 'default_uaf - change default record'
;  2437     2517  1 global routine default_uaf : novalue =
;  2438     2518  2 begin
;  2439     2519  2
;  2440     2520  2 !++
;  2441     2521  2 !
;  2442     2522  2 ! FUNCTIONAL DESCRIPTION:
;  2443     2523  2 !
;  2444     2524  2 !       Change a default value in the default record.
;  2445     2525  2 !
;  2446     2526  2 ! INPUTS:
;  2447     2527  2 !
;  2448     2528  2 !       none
;  2449     2529  2 !
;  2450     2530  2 ! IMPLICIT INPUTS:
;  2451     2531  2 !
;  2452     2532  2 !       none
;  2453     2533  2 !
;  2454     2534  2 ! OUTPUTS:
;  2455     2535  2 !
;  2456     2536  2 !       none
;  2457     2537  2 !
;  2458     2538  2 ! IMPLICIT OUTPUTS:
;  2459     2539  2 !
;  2460     2540  2 !       none
;  2461     2541  2 !
;  2462     2542  2 ! ROUTINE VALUE:
;  2463     2543  2 !
;  2464     2544  2 !       none
;  2465     2545  2 !
;  2466     2546  2 ! SIDE EFFECTS:
;  2467     2547  2 !
;  2468     2548  2 !       none
;  2469     2549  2 !--
;  2470     2550  2
;  2471     2551  2
;  2472     2552  2 !
;  2473     2553  2 ! Locate default record and load it into RECBUF
;  2474     2554  2 ! if not already there (via an indirect MODIFY DEFAULT)
;  2475     2555  2 !
;  2476     2556  2
;  2477     2557  2 if not .mod_default
;  2478     2558  2 then
;  2479     2559  3     begin
;  2480     2560  3     if not locate_user (.defuser<0,8>, defuser+1, true)
;  2481     2561  3     then
;  2482     2562  4         begin
;  2483     2563  4         LIB$SIGNAL(UAF$_DEFERR, 0, .rmserr);
;  2484     2564  4         return;
;  2485     2565  3         end;
;  2486     2566  2     end;
;  2487     2567  2 !
;  2488     2568  2 ! The encrypted password field of the DEFAULT record can not be propagated
;  2489     2569  2 ! to another user, because the encryption algorithm takes the user name as
;  2490     2570  2 ! an input.  The user is merely warned that this qualifier has no effect.
;  2491     2571  2 !
;  2492     2572  2
```

```
;  2493          2573  2 pwd_flag = true;
;  2494          2574  2
;  2495          2575  2 !
;  2496          2576  2 ! Update values supplied and exit if errors.
;  2497          2577  2 !
;  2498          2578  2
;  2499          2579  2 if update_record ()
;  2500          2580  2 then
;  2501          2581  3     begin
;  2502          2582  3     if not .pwd_flag
;  2503          2583  3     then LIB$SIGNAL(UAF$_NODEFPWD);
;  2504          2584  3
;  2505          2585  3 !
;  2506          2586  3 ! Now write the modified record back into the DEFAULT_RECORD buffer.
;  2507          2587  3 !
;  2508          2588  3     default_size = .uafrab[rab$w_rsz];
;  2509          2589  3     ch$move (.default_size, recbuf, default_record);
;  2510          2590  3
;  2511          2591  3 !
;  2512          2592  3 ! Update the default record in the file.  Note that file has changed.
;  2513          2593  3 !
;  2514          2594  4     if not rmsok ($update (rab = uafrab))
;  2515          2595  3     then
;  2516          2596  3         LIB$SIGNAL(UAF$_MDFYERR, 0, .rmserr)
;  2517          2597  3     else
;  2518          2598  4         begin
;  2519          2599  4         modify_flag = true;
;  2520          2600  4         security_audit (nsa$k_recid_sysuaf_mod);
;  2521          2601  4         if not .mod_default
;  2522          2602  4         then
;  2523          2603  4             LIB$SIGNAL(UAF$_MDFYMSG)
;  2524          2604  4         else
;  2525          2605  4             mod_default = false;
;  2526          2606  3         end;
;  2527          2607  3     end
;  2528          2608  3
;  2529          2609  2 else
;  2530          2610  2     $release (rab = uafrab);                    ! unlock the record
;  2531          2611  2
;  2532          2612  1 end;


                                          01FC 00000          .ENTRY  DEFAULT_UAF, Save R2,R3,R4,R5,R6,R7,R8      ; 2517
                                58 00000000G   00   9E 00002  MOVAB   LIB$SIGNAL, R8                              ;
                                57 00000000'   00   9E 00009  MOVAB   MOD_DEFAULT, R7                            ;
                                56 00000000'   00   9E 00010  MOVAB   RMSERR, R6                                 ;
                                25             67   E8 00017  BLBS    MOD_DEFAULT, 1$                           ; 2557
                                               01   DD 0001A  PUSHL   #1                                        ; 2560
                                   00000000'   00   9F 0001C  PUSHAB  DEFUSER+1                                  ;
                                7E 00000000'   00   9A 00022  MOVZBL  DEFUSER, -(SP)                           ;
                          00000000V 00         03   FB 00029  CALLS   #3, LOCATE_USER                           ;
                                0C             50   E8 00030  BLBS    R0, 1$                                    ;
                                               66   DD 00033  PUSHL   RMSERR                                    ; 2563
                                               7E   D4 00035  CLRL    -(SP)                                     ;
```

```
                                00000000G  8F   DD 00037          PUSHL   #UAF$_DEFERR                                    ;
                                           48   11 0003D          BRB     3$                                              ;
                          0680  C6         01   D0 0003F 1$:      MOVL    #1, PWD_FLAG                                    ; 2573
                00000000G  00              00   FB 00044          CALLS   #0, UPDATE_RECORD                               ; 2579
                           5E              50   E9 0004B          BLBC    R0, 6$                                          ;
                           09        0680  C6   E8 0004E          BLBS    PWD_FLAG, 2$                                    ; 2582
                                00000000G  8F   DD 00053          PUSHL   #UAF$_NODEFPWD                                  ; 2583
                           68              01   FB 00059          CALLS   #1, LIB$SIGNAL                                  ;
                          0458  C7   0612  C6   B0 0005C 2$:      MOVW    UAFRAB+34, DEFAULT_SIZE                         ; 2588
              045C  C7      08  A6   0458  C7   28 00063          MOVC3   DEFAULT_SIZE, RECBUF, DEFAULT_RECORD            ; 2589
                                05F0  C6   9F 0006C          PUSHAB  UAFRAB                                          ; 2594
                00000000G  00              01   FB 00070          CALLS   #1, SYS$UPDATE                                  ;
                           66              50   D0 00077          MOVL    R0, RMSERR                                      ;
                           0E              50   E8 0007A          BLBS    R0, 4$                                          ;
                                           66   DD 0007D          PUSHL   RMSERR                                          ; 2596
                                           7E   D4 0007F          CLRL    -(SP)                                           ;
                                00000000G  8F   DD 00081          PUSHL   #UAF$_MDFYERR                                   ;
                           68              03   FB 00087 3$:      CALLS   #3, LIB$SIGNAL                                  ;
                                           04   0008A            RET                                                     ;
                          04   A7         01   D0 0008B 4$:      MOVL    #1, MODIFY_FLAG                                 ; 2599
                                00030002   8F   DD 0008F          PUSHL   #196610                                         ; 2600
                00000000V  00              01   FB 00095          CALLS   #1, SECURITY_AUDIT                              ;
                           0A              67   E8 0009C          BLBS    MOD_DEFAULT, 5$                                 ; 2601
                                00000000G  8F   DD 0009F          PUSHL   #UAF$_MDFYMSG                                   ; 2603
                           68              01   FB 000A5          CALLS   #1, LIB$SIGNAL                                  ;
                                           04   000A8            RET                                                     ;
                                           67   D4 000A9 5$:      CLRL    MOD_DEFAULT                                     ; 2605
                                           04   000AB            RET                                                     ; 2579
                                05F0  C6   9F 000AC 6$:      PUSHAB  UAFRAB                                          ; 2610
                00000000G  00              01   FB 000B0          CALLS   #1, SYS$RELEASE                                 ;
                                           04   000B7            RET                                                     ; 2612

; Routine Size:  184 bytes,    Routine Base:  $CODE$ + 0E64
```

```
; 2534          2613  1 %sbttl 'list_proxy - list proxy entries in NETUAF.LIS'
; 2535          2614  1 global routine list_proxy : novalue =
; 2536          2615  1 begin
; 2537          2616  2
; 2538          2617  2 !++
; 2539          2618  2 !
; 2540          2619  2 ! FUNCTIONAL DESCRIPTION:
; 2541          2620  2 !
; 2542          2621  2 !       This routine produces a listing of the entire NETUAF.DAT file
; 2543          2622  2 !       in NETUAF.LIS.
; 2544          2623  2 !
; 2545          2624  2 ! INPUTS:
; 2546          2625  2 !
; 2547          2626  2 !       none
; 2548          2627  2 !
; 2549          2628  2 ! OUTPUTS:
; 2550          2629  2 !
; 2551          2630  2 !       none
; 2552          2631  2 !
; 2553          2632  2 ! SIDE EFFECTS:
; 2554          2633  2 !
; 2555          2634  2 !       A listing file is produced
; 2556          2635  2 !
; 2557          2636  2 !--
; 2558          2637  2
; 2559          2638  2 local
; 2560          2639  2         action;
; 2561          2640  2
; 2562          2641  2 !
; 2563          2642  2 ! Make sure NETUAF.DAT exists
; 2564          2643  2 !
; 2565          2644  2 if not .netuaf_exists
; 2566          2645  2 then return LIB$SIGNAL(UAF$_NAFDNE);
; 2567          2646  2
; 2568          2647  2 !
; 2569          2648  2 ! Set up the listing file FAB and connect RAB
; 2570          2649  2 !
; 2571          2650  2 nlstfab[fab$v_dlt] = false;
; 2572          2651  3 if rmsbad ($create (fab = nlstfab))
; 2573          2652  2 then return LIB$SIGNAL(UAF$_LSTERR, 0, .rmserr);
; 2574          2653  2
; 2575          2654  3 if rmsbad ($connect (rab = nlstrab))
; 2576          2655  2 then return LIB$SIGNAL(UAF$_LSTERR, 0, .rmserr);
; 2577          2656  2
; 2578          2657  2 !
; 2579          2658  2 ! Set action routine and rab pointer
; 2580          2659  2 !
; 2581          2660  2 wild_netuser = true;
; 2582          2661  2 match_tokenlen = 1;
; 2583          2662  2 match_token = %c'*';
; 2584          2663  2 rabptr = nlstrab;
; 2585          2664  2 action = display_proxy;
; 2586          2665  2 header_flag = true;
; 2587          2666  2 nafrab[rab$l_rop] = rab$m_rrl or rab$m_nlk;
; 2588          2667  2
; 2589          2668  2 LIB$SIGNAL(UAF$_LSTMSG1);
; 2590          2669  2
```

```
;  2591          2670  2 !
;  2592          2671  2 ! LOCATE_PROXY will call the action routine for each proxy record
;  2593          2672  2 !
;  2594          2673  3 if rmsbad (locate_proxy (.action))
;  2595          2674  2 then
;  2596          2675  3     begin
;  2597          2676  3     if .rmserr eql rms$_rnf
;  2598          2677  3     then
;  2599          2678  3         LIB$SIGNAL(UAF$_BADSPC)
;  2600          2679  3     else
;  2601          2680  3         LIB$SIGNAL(UAF$_LSTERR, 0, .rmserr);
;  2602          2681  3     end
;  2603          2682  2 else
;  2604          2683  2     LIB$SIGNAL(UAF$_NETLSTMSG);
;  2605          2684  2
;  2606          2685  2 !
;  2607          2686  2 ! Disconnect RAB and close FAB
;  2608          2687  2 !
;  2609          2688  2 $disconnect (rab = nlstrab);
;  2610          2689  2 $close (fab = nlstfab);
;  2611          2690  2
;  2612          2691  1 end;


                                                        .EXTRN  SYS$DISCONNECT, SYS$CLOSE

                                        007C 00000      .ENTRY  LIST_PROXY, Save R2,R3,R4,R5,R6                 ; 2614
                    56 00000000G    8F  D0 00002        MOVL    #UAF$_LSTERR, R6
                    55 00000000G    00  9E 00009        MOVAB   LIB$SIGNAL, R5
                    54 00000000'    00  9E 00010        MOVAB   NLSTRAB, R4
                    53 00000000'    00  9E 00017        MOVAB   RMSERR, R3
                    0A        F8    A3  E8 0001E        BLBS    NETUAF_EXISTS, 1$                               ; 2644
                       00000000G    8F  DD 00022        PUSHL   #UAF$_NAFDNE                                    ; 2645
                    65              01  FB 00028        CALLS   #1, LIB$SIGNAL
                                    04  0002B        RET
                B5  A4        80    8F  8A 0002C 1$:    BICB2   #128, NLSTFAB+5                                 ; 2650
                              B0    A4  9F 00031        PUSHAB  NLSTFAB                                         ; 2651
00000000G       00                  01  FB 00034        CALLS   #1, SYS$CREATE
                63                  50  D0 0003B        MOVL    R0, RMSERR
                0F                  50  E9 0003E        BLBC    R0, 2$
                                    54  DD 00041        PUSHL   R4                                              ; 2654
00000000G       00                  01  FB 00043        CALLS   #1, SYS$CONNECT
                63                  50  D0 0004A        MOVL    R0, RMSERR
                0A                  50  E8 0004D        BLBS    R0, 3$
                                    63  DD 00050 2$:    PUSHL   RMSERR                                          ; 2655
                                    7E  D4 00052        CLRL    -(SP)
                                    56  DD 00054        PUSHL   R6
                    65              03  FB 00056        CALLS   #3, LIB$SIGNAL
                                    04  00059        RET
                E0  A3              01  D0 0005A 3$:    MOVL    #1, WILD_NETUSER                                ; 2660
                DC  A3              01  D0 0005E        MOVL    #1, MATCH_TOKENLEN                              ; 2661
                98  A3              2A  D0 00062        MOVL    #42, MATCH_TOKEN                                ; 2662
                80  A3              64  9E 00066        MOVAB   NLSTRAB, RABPTR                                 ; 2663
                    52 00000000V    00  9E 0006A        MOVAB   DISPLAY_PROXY, ACTION                           ; 2664
              FCC4  C4              01  D0 00071        MOVL    #1, HEADER_FLAG                                 ; 2665
              0638  C3 00100008     8F  D0 00076        MOVL    #1048584, NAFRAB+4                              ; 2666
```

```
                                 00000000G   8F   DD 0007F          PUSHL   #UAF$_LSTMSG1                           ; 2668
                                       65    01   FB 00085          CALLS   #1, LIB$SIGNAL
                                             52   DD 00088          PUSHL   ACTION                                 ; 2673
                          00000000V  00      01   FB 0008A          CALLS   #1, LOCATE_PROXY
                                     63      50   D0 00091          MOVL    R0, RMSERR
                                     1F      50   E8 00094          BLBS    R0, 5$
                                     50      63   D0 00097          MOVL    RMSERR, R0                             ; 2676
                          000182B2   8F      50   D1 0009A          CMPL    R0, #98994
                                             08   12 000A1          BNEQ    4$
                                 00000000G   8F   DD 000A3          PUSHL   #UAF$_BADSPC                           ; 2678
                                             11   11 000A9          BRB     6$
                                             50   DD 000AB  4$:     PUSHL   R0                                     ; 2680
                                             7E   D4 000AD          CLRL    -(SP)
                                             56   DD 000AF          PUSHL   R6
                                       65    03   FB 000B1          CALLS   #3, LIB$SIGNAL
                                             09   11 000B4          BRB     7$                                     ; 2673
                                 00000000G   8F   DD 000B6  5$:     PUSHL   #UAF$_NETLSTMSG                        ; 2683
                                       65    01   FB 000BC  6$:     CALLS   #1, LIB$SIGNAL
                                             54   DD 000BF  7$:     PUSHL   R4                                     ; 2688
                          00000000G  00      01   FB 000C1          CALLS   #1, SYS$DISCONNECT
                                        B0   A4   9F 000C8          PUSHAB  NLSTFAB                                ; 2689
                          00000000G  00      01   FB 000CB          CALLS   #1, SYS$CLOSE
                                             04   000D2            RET                                            ; 2691

; Routine Size:  211 bytes,    Routine Base:  $CODE$ + 0F1C
```

```
;  2614     2692  1 %sbttl 'list_uaf - list file routine'
;  2615     2693  1 global routine list_uaf : novalue =
;  2616     2694  2 begin
;  2617     2695  2
;  2618     2696  2 !++
;  2619     2697  2 !
;  2620     2698  2 ! FUNCTIONAL DESCRIPTION:
;  2621     2699  2 !
;  2622     2700  2 !       Display the specified users in a file named 'SYSUAF.LIS'.
;  2623     2701  2 !
;  2624     2702  2 ! INPUTS:
;  2625     2703  2 !
;  2626     2704  2 !       none
;  2627     2705  2 !
;  2628     2706  2 ! IMPLICIT INPUTS:
;  2629     2707  2 !
;  2630     2708  2 !       none
;  2631     2709  2 !
;  2632     2710  2 ! OUTPUTS:
;  2633     2711  2 !
;  2634     2712  2 !       none
;  2635     2713  2 !
;  2636     2714  2 ! IMPLICIT OUTPUTS:
;  2637     2715  2 !
;  2638     2716  2 !       none
;  2639     2717  2 !
;  2640     2718  2 ! ROUTINE VALUE:
;  2641     2719  2 !
;  2642     2720  2 !       none
;  2643     2721  2 !
;  2644     2722  2 ! SIDE EFFECTS:
;  2645     2723  2 !
;  2646     2724  2 !       none
;  2647     2725  2 !--
;  2648     2726  2
;  2649     2727  2 local
;  2650     2728  2         action;
;  2651     2729  2
;  2652     2730  2 !
;  2653     2731  2 ! Obtain the user specification.  This sets wildcard flags and initializes
;  2654     2732  2 ! the appropriate key in RECBUF.
;  2655     2733  2 !
;  2656     2734  2
;  2657     2735  2 if not parse_wild (sd_token1,true)              ! Null string defaults to *
;  2658     2736  2 then return;
;  2659     2737  2
;  2660     2738  2 !
;  2661     2739  2 ! Obtain qualifiers.  This determines which display should be used.
;  2662     2740  2 !
;  2663     2741  2 full_flag = false;
;  2664     2742  2 brief_flag = true;
;  2665     2743  2
;  2666     2744  3 if cli$present (sd_full) or (not cli$present (sd_brief))
;  2667     2745  2 then
;  2668     2746  3     begin
;  2669     2747  3     brief_flag = false;
;  2670     2748  3     full_flag  = true;
```

```
;  2671     2749  2     end;
;  2672     2750  2
;  2673     2751  2 !
;  2674     2752  2 ! Create the listing file.
;  2675     2753  2 !
;  2676     2754  2
;  2677     2755  2 lstfab[fab$v_dlt] = false;              ! initialize DLT bit
;  2678     2756  3 if rmsbad ($create (fab = lstfab))
;  2679     2757  2 then return LIB$SIGNAL(UAF$_LSTERR, 0, .rmserr);
;  2680     2758  2
;  2681     2759  3 if rmsbad ($connect (rab = lstrab))
;  2682     2760  2 then return LIB$SIGNAL(UAF$_LSTERR, 0, .rmserr);
;  2683     2761  2
;  2684     2762  2 !
;  2685     2763  2 ! Request a header record for the file and aim RABPTR at our RAB.
;  2686     2764  2 !
;  2687     2765  2
;  2688     2766  2 header_flag = true;
;  2689     2767  2 rabptr = lstrab;
;  2690     2768  2 found_match = false;
;  2691     2769  2 uafrab[rab$l_rop] = rab$m_rrl or rab$m_nlk;
;  2692     2770  2 !
;  2693     2771  2 ! Flag the list operation for the rights data base routines .
;  2694     2772  2 !
;  2695     2773  2 rdb_list_flag = true ;
;  2696     2774  2
;  2697     2775  2 !
;  2698     2776  2 ! Choose the appropriate display.
;  2699     2777  2 !
;  2700     2778  2
;  2701     2779  2 action = display_brief;
;  2702     2780  2 if .full_flag
;  2703     2781  2 then action = display_full;
;  2704     2782  2
;  2705     2783  2 LIB$SIGNAL(UAF$_LSTMSG1);                    ! announce starting
;  2706     2784  2
;  2707     2785  3 if rmsbad (wild_user (.action))
;  2708     2786  2 then
;  2709     2787  3     begin
;  2710     2788  3     if .rmserr eql rms$_rnf
;  2711     2789  3     then
;  2712     2790  3         LIB$SIGNAL(UAF$_BADSPC)
;  2713     2791  3     else
;  2714     2792  3         LIB$SIGNAL(UAF$_LSTERR, 0, .rmserr);
;  2715     2793  3     lstfab[fab$v_dlt] = true;            ! press delete button
;  2716     2794  3     end
;  2717     2795  2 else
;  2718     2796  2     LIB$SIGNAL(UAF$_LSTMSG2);
;  2719     2797  2
;  2720     2798  2 $disconnect (rab = lstrab);
;  2721     2799  2 $close (fab = lstfab);
;  2722     2800  1 end;
```

```
                                 01FC 00000          .ENTRY  LIST_UAF, Save R2,R3,R4,R5,R6,R7,R8                 ; 2693
                          58 00000000G   8F   D0 00002       MOVL    #UAF$_LSTERR, R8                            ;
                          57 00000000G   00   9E 00009       MOVAB   CLI$PRESENT, R7                             ;
                          56 00000000'   00   9E 00010       MOVAB   SD_TOKEN1, R6                               ;
                          55 00000000G   00   9E 00017       MOVAB   LIB$SIGNAL, R5                              ;
                          54 00000000'   00   9E 0001E       MOVAB   RMSERR, R4                                  ;
                          53 00000000'   00   9E 00025       MOVAB   LSTRAB, R3                                  ;
                                         01   DD 0002C       PUSHL   #1                                          ; 2735
                                         56   DD 0002E       PUSHL   R6                                          ;
                    00000000G     00         02   FB 00030       CALLS   #2, PARSE_WILD                          ;
                                  01         50   E8 00037       BLBS    R0, 1$                                  ;
                                             04   00003A       RET                                             ;
                        FE08      C3         01   7D 0003B 1$:  MOVQ    #1, BRIEF_FLAG                          ; 2742
                                       2C    A6   9F 00040       PUSHAB  SD_FULL                                 ; 2744
                                  67         01   FB 00043       CALLS   #1, CLI$PRESENT                         ;
                                  09         50   E8 00046       BLBS    R0, 2$                                  ;
                                       20    A6   9F 00049       PUSHAB  SD_BRIEF                                ;
                                  67         01   FB 0004C       CALLS   #1, CLI$PRESENT                         ;                  20
                                  09         50   E8 0004F       BLBS    R0, 3$                                  ;                  20
                                        FE08 C3   D4 00052 2$:    CLRL    BRIEF_FLAG                              ; 2747
                          FE0C    C3         01   D0 00056       MOVL    #1, FULL_FLAG                           ; 2748
                            B5    A3         80   8F 8A 0005B 3$:  BICB2   #128, LSTFAB+5                       ; 2755             20
                                       B0    A3   9F 00060       PUSHAB  LSTFAB                                  ; 2756
                    00000000G     00         01   FB 00063       CALLS   #1, SYS$CREATE                          ;
                                  64         50   D0 0006A       MOVL    R0, RMSERR                              ;
                                  0F         50   E9 0006D       BLBC    R0, 4$                                  ;
                                             53   DD 00070       PUSHL   R3                                      ; 2759
                    00000000G     00         01   FB 00072       CALLS   #1, SYS$CONNECT                         ;
                                  64         50   D0 00079       MOVL    R0, RMSERR                              ;
                                  0A         50   E8 0007C       BLBS    R0, 5$                                  ;
                                             64   DD 0007F 4$:  PUSHL   RMSERR                                  ; 2760
                                             7E   D4 00081       CLRL    -(SP)                                   ;
                                             58   DD 00083       PUSHL   R8                                      ;
                                  65         03   FB 00085       CALLS   #3, LIB$SIGNAL                          ;
                                             04   00088       RET                                             ;
                        FE10      C3         01   D0 00089 5$:  MOVL    #1, HEADER_FLAG                         ; 2766
                          80      A4         63   9E 0008E       MOVAB   LSTRAB, RABPTR                          ; 2767
                                       06C8  C4   D4 00092       CLRL    FOUND_MATCH                             ; 2768
                        05F4      C4 00100008  8F   D0 00096       MOVL    #1048584, UAFRAB+4                      ; 2769
                    00000000G     00         01   90 0009F       MOVB    #1, RDB_LIST_FLAG                       ; 2773
                                  52 00000000V 00   9E 000A6       MOVAB   DISPLAY_BRIEF, ACTION                   ; 2779
                                  07      FE0C C3   E9 000AD       BLBC    FULL_FLAG, 6$                           ; 2780
                                  52 00000000V 00   9E 000B2       MOVAB   DISPLAY_FULL, ACTION                    ; 2781
                                     00000000G 8F   DD 000B9 6$:   PUSHL   #UAF$_LSTMSG1                          ; 2783
                                  65         01   FB 000BF       CALLS   #1, LIB$SIGNAL                          ;
                                             52   DD 000C2       PUSHL   ACTION                                  ; 2785
                    00000000V     00         01   FB 000C4       CALLS   #1, WILD_USER                           ;
                                  64         50   D0 000CB       MOVL    R0, RMSERR                              ;
                                  27         50   E8 000CE       BLBS    R0, 9$                                  ;
                                  50         64   D0 000D1       MOVL    RMSERR, R0                              ; 2788
                    000182B2      8F         50   D1 000D4       CMPL    R0, #98994                              ;
                                             0B   12 000DB       BNEQ    7$                                      ;
                                     00000000G 8F   DD 000DD       PUSHL   #UAF$_BADSPC                            ; 2790
                                  65         01   FB 000E3       CALLS   #1, LIB$SIGNAL                          ;
                                             09   11 000E6       BRB     8$                                      ;
                                             50   DD 000E8 7$:  PUSHL   R0                                      ; 2792
                                             7E   D4 000EA       CLRL    -(SP)                                   ;
```

```
                                                58  DD 000EC          PUSHL   R8                                          ;
                                          65    03  FB 000EE          CALLS   #3, LIB$SIGNAL                              ;
                                    B5    A3        80  8F  88 000F1 8$:    BISB2   #128, LSTFAB+5                      ; 2793
                                                      09  11 000F6          BRB     10$                                 ; 2785
                                          00000000G   8F  DD 000F8 9$:    PUSHL   #UAF$_LSTMSG2                       ; 2796
                                          65          01  FB 000FE          CALLS   #1, LIB$SIGNAL                      ;
                                                      53  DD 00101 10$:    PUSHL   R3                                  ; 2798
                                 00000000G  00        01  FB 00103          CALLS   #1, SYS$DISCONNECT                  ;
                                                B0    A3  9F 0010A          PUSHAB  LSTFAB                              ; 2799
                                 00000000G  00        01  FB 0010D          CALLS   #1, SYS$CLOSE                       ;
                                                      04  00114          RET                                         ; 2800

; Routine Size:  277 bytes,    Routine Base:  $CODE$ + 0FEF
```

```
;  2724     2801  1 %sbttl 'show_user_uaf - display user record'
;  2725     2802  1 global routine show_user_uaf : novalue =
;  2726     2803  2 begin
;  2727     2804  2
;  2728     2805  2 !++
;  2729     2806  2 !
;  2730     2807  2 ! FUNCTIONAL DESCRIPTION:
;  2731     2808  2 !
;  2732     2809  2 !       Display the specified users on SYS$OUTPUT.
;  2733     2810  2 !
;  2734     2811  2 ! INPUTS:
;  2735     2812  2 !
;  2736     2813  2 !       none
;  2737     2814  2 !
;  2738     2815  2 ! IMPLICIT INPUTS:
;  2739     2816  2 !
;  2740     2817  2 !       none
;  2741     2818  2 !
;  2742     2819  2 ! OUTPUTS:
;  2743     2820  2 !
;  2744     2821  2 !       none
;  2745     2822  2 !
;  2746     2823  2 ! IMPLICIT OUTPUTS:
;  2747     2824  2 !
;  2748     2825  2 !       none
;  2749     2826  2 !
;  2750     2827  2 ! ROUTINE VALUE:
;  2751     2828  2 !
;  2752     2829  2 !       none
;  2753     2830  2 !
;  2754     2831  2 ! SIDE EFFECTS:
;  2755     2832  2 !
;  2756     2833  2 !       none
;  2757     2834  2 !--
;  2758     2835  2
;  2759     2836  2 local
;  2760     2837  2         action;
;  2761     2838  2
;  2762     2839  2 !
;  2763     2840  2 ! Obtain the user specification.  This sets wildcard flags and initializes
;  2764     2841  2 ! the appropriate key in RECBUF.
;  2765     2842  2 !
;  2766     2843  2
;  2767     2844  2 if not parse_wild (sd_token1,false)             ! Null string is disallowed.
;  2768     2845  2 then return;
;  2769     2846  2
;  2770     2847  2 !
;  2771     2848  2 ! Obtain qualifiers.  This determines which display should be used.
;  2772     2849  2 !
;  2773     2850  2 full_flag = true;
;  2774     2851  2 brief_flag = false;
;  2775     2852  2
;  2776     2853  3 if cli$present (sd_brief) or (not cli$present (sd_full))
;  2777     2854  2 then
;  2778     2855  3     begin
;  2779     2856  3     brief_flag = true;
;  2780     2857  3     full_flag  = false;
```

```
;  2781          2858  2     end;
;  2782          2859  2
;  2783          2860  2 !
;  2784          2861  2 ! Request a header record for the file and aim RABPTR at our RAB.
;  2785          2862  2 !
;  2786          2863  2
;  2787          2864  2 header_flag = true;
;  2788          2865  2 rabptr = outrab;
;  2789          2866  2 found_match = false;
;  2790          2867  2 uafrab[rab$l_rop] = rab$m_rrl or rab$m_nlk;
;  2791          2868  2
;  2792          2869  2 !
;  2793          2870  2 ! Flag the show operation for the rights data base routines .
;  2794          2871  2 !
;  2795          2872  2 rdb_list_flag = false ;
;  2796          2873  2
;  2797          2874  2 !
;  2798          2875  2 ! Choose the appropriate display.
;  2799          2876  2 !
;  2800          2877  2
;  2801          2878  2 action = display_full;
;  2802          2879  2 if .brief_flag
;  2803          2880  2 then action = display_brief;
;  2804          2881  2
;  2805          2882  3 if rmsbad (wild_user (.action))
;  2806          2883  2 then
;  2807          2884  3     begin
;  2808          2885  3     if .rmserr eql rms$_rnf
;  2809          2886  3     then
;  2810          2887  3         LIB$SIGNAL(UAF$_BADSPC)
;  2811          2888  3     else
;  2812          2889  3         LIB$SIGNAL(UAF$_SHOW_ERR, 0, .rmserr);
;  2813          2890  2     end;
;  2814          2891  1 end;


                                                007C 00000          .ENTRY  SHOW_USER_UAF, Save R2,R3,R4,R5,R6      ; 2802
                                    56 00000000G   00   9E 00002    MOVAB   LIB$SIGNAL, R6                          ;
                                    55 00000000G   00   9E 00009    MOVAB   CLI$PRESENT, R5                         ;
                                    54 00000000'   00   9E 00010    MOVAB   SD_TOKEN1, R4                           ;
                                    53 00000000'   00   9E 00017    MOVAB   BRIEF_FLAG, R3                          ;
                                    52 00000000'   00   9E 0001E    MOVAB   RMSERR, R2                              ;
                                                   7E   D4 00025    CLRL    -(SP)                                   ; 2844
                                                   54   DD 00027    PUSHL   R4                                      ;
                          00000000G   00           02   FB 00029    CALLS   #2, PARSE_WILD                          ;
                                      7B           50   E9 00030    BLBC    R0, 5$                                  ;
                                04    A3           01   D0 00033    MOVL    #1, FULL_FLAG                           ; 2850
                                                   63   D4 00037    CLRL    BRIEF_FLAG                              ; 2851
                                                20 A4   9F 00039    PUSHAB  SD_BRIEF                                ; 2853
                                      65           01   FB 0003C    CALLS   #1, CLI$PRESENT                         ;
                                      09           50   E8 0003F    BLBS    R0, 1$                                  ;
                                                2C A4   9F 00042    PUSHAB  SD_FULL                                 ;
                                      65           01   FB 00045    CALLS   #1, CLI$PRESENT                         ;
                                      03           50   E8 00048    BLBS    R0, 2$                                  ;
```

```
                                63              01  7D 0004B 1$:    MOVQ    #1, BRIEF_FLAG                  ; 2856
                             08 A3              01  D0 0004E 2$:    MOVL    #1, HEADER_FLAG                 ; 2864
                             80 A2      0684    CE  9E 00052        MOVAB   OUTRAB, RABPTR                  ; 2865
                                        06C8    C2  D4 00058        CLRL    FOUND_MATCH                     ; 2866
                           05F4 C2 00100008     8F  D0 0005C        MOVL    #1048584, UAFRAB+4              ; 2867
                                   00000000G    00  94 00065        CLRB    RDB_LIST_FLAG                   ; 2872
                                50 00000000V    00  9E 00068        MOVAB   DISPLAY_FULL, ACTION            ; 2878
                                07              63  E9 00072        BLBC    BRIEF_FLAG, 3$                  ; 2879
                                50 00000000V    00  9E 00075        MOVAB   DISPLAY_BRIEF, ACTION           ; 2880
                                                50  DD 0007C 3$:    PUSHL   ACTION                          ; 2882
                    00000000V   00              01  FB 0007E        CALLS   #1, WILD_USER                   ;
                                62              50  D0 00085        MOVL    RO, RMSERR                      ;
                                23              50  E8 00088        BLBS    RO, 5$                          ;
                                50              62  D0 0008B        MOVL    RMSERR, RO                      ; 2885
                    000182B2    8F              50  D1 0008E        CMPL    RO, #98994                      ;
                                                0A  12 00095        BNEQ    4$                              ;
                                   00000000G    8F  DD 00097        PUSHL   #UAF$_BADSPC                    ; 2887
                                66              01  FB 0009D        CALLS   #1, LIB$SIGNAL                  ;
                                                04  00A0          RET                                     ;
                                                50  DD 000A1 4$:    PUSHL   RO                              ; 2889
                                                7E  D4 000A3        CLRL    -(SP)                           ;
                                   00000000G    8F  DD 000A5        PUSHL   #UAF$_SHOW_ERR                  ;
                                66              03  FB 000AB        CALLS   #3, LIB$SIGNAL                  ;
                                                04  00AE 5$:    RET                                     ; 2891

; Routine Size:  175 bytes,    Routine Base:  $CODE$ + 1104
```

```
;2816     2892  1 %sbttl 'show_proxy - display proxy record at terminal'
;2817     2893  1 global routine show_proxy : novalue =
;2818     2894  2 begin
;2819     2895  2
;2820     2896  2 !++
;2821     2897  2 !
;2822     2898  2 ! FUNCTIONAL DESCRIPTION:
;2823     2899  2 !
;2824     2900  2 !       This routine will display a specific proxy record or
;2825     2901  2 !       will display all proxy entries to the user terminal.
;2826     2902  2 !
;2827     2903  2 ! INPUTS:
;2828     2904  2 !
;2829     2905  2 !       none
;2830     2906  2 !
;2831     2907  2 ! OUTPUTS:
;2832     2908  2 !
;2833     2909  2 !       none
;2834     2910  2 !
;2835     2911  2 ! IMPLICIT INPUTS:
;2836     2912  2 !
;2837     2913  2 !       TOKENLEN, TOKENPTR
;2838     2914  2 !
;2839     2915  2 ! IMPLICIT OUTPUTS:
;2840     2916  2 !
;2841     2917  2 !       none
;2842     2918  2 !
;2843     2919  2 ! SIDE EFFECTS:
;2844     2920  2 !
;2845     2921  2 !       none
;2846     2922  2 !
;2847     2923  2 !--
;2848     2924  2
;2849     2925  2 local
;2850     2926  2         node_len,
;2851     2927  2         node_ptr,
;2852     2928  2         remuser_len,
;2853     2929  2         remuser_ptr,
;2854     2930  2         action,
;2855     2931  2         counter,
;2856     2932  2         success;
;2857     2933  2
;2858     2934  2 !
;2859     2935  2 ! Make sure that NETUAF.DAT exists
;2860     2936  2 !
;2861     2937  2 if not .netuaf_exists
;2862     2938  2 then return LIB$SIGNAL(UAF$_NAFDNE);
;2863     2939  2
;2864     2940  2 !
;2865     2941  2 ! Retrieve token
;2866     2942  2 !
;2867     2943  2 cli$get_value (sd_token1, tokendsc);
;2868     2944  2
;2869     2945  2 header_flag = true;
;2870     2946  2
;2871     2947  2 !
;2872     2948  2 ! Wildcard spec?
```

```
;  2873          2949  2 !
;  2874          2950  2 if ch$find_ch (.tokenlen, .tokenptr, %c'%') neq 0
;  2875          2951  2 or ch$find_ch (.tokenlen, .tokenptr, %c'*') neq 0
;  2876          2952  2 then
;  2877          2953  3     begin
;  2878          2954  3     wild_netuser = true;
;  2879          2955  3     match_tokenlen = .tokenlen;
;  2880          2956  3     ch$move (.tokenlen, .tokenptr, match_token);
;  2881          2957  3     end
;  2882          2958  3 !
;  2883          2959  3 ! Otherwise, just display a single entry
;  2884          2960  3 !
;  2885          2961  2 else
;  2886          2962  3     begin
;  2887          2963  3     wild_netuser = false;
;  2888          2964  3     if not remote_parse (node_ptr, node_len, remuser_ptr, remuser_len)
;  2889          2965  3     then return;
;  2890          2966  3
;  2891          2967  3     ch$copy (.node_len, .node_ptr, ' ', naf$s_node, netbuf[naf$t_node]);
;  2892          2968  3     ch$copy (.remuser_len, .remuser_ptr, ' ',
;  2893          2969  3              naf$s_remuser, netbuf[naf$t_remuser]);
;  2894          2970  2     end;
;  2895          2971  2
;  2896          2972  2 !
;  2897          2973  2 ! Set up action routine and rab pointer
;  2898          2974  2 !
;  2899          2975  2 rabptr = outrab;
;  2900          2976  2 action = display_proxy;
;  2901          2977  2 found_match = false;
;  2902          2978  2 nafrab[rab$l_rop] = rab$m_rrl or rab$m_nlk;
;  2903          2979  2
;  2904          2980  2 !
;  2905          2981  2 ! Make call (s) necessary to display the rquested entry or entries
;  2906          2982  2 !
;  2907          2983  3 if rmsbad (locate_proxy (.action))
;  2908          2984  2 then
;  2909          2985  3     begin
;  2910          2986  3     if .rmserr eql rms$_rnf
;  2911          2987  3     then
;  2912          2988  3         LIB$SIGNAL(UAF$_BADSPC)
;  2913          2989  3     else
;  2914          2990  3         LIB$SIGNAL(UAF$_SHOW_ERR, 0, .rmserr);
;  2915          2991  2     end;
;  2916          2992  1 end;


                                00FC 00000          .ENTRY  SHOW_PROXY, Save R2,R3,R4,R5,R6,R7          ; 2893
                57 00000000G    00   9E 00002          MOVAB   LIB$SIGNAL, R7                              ;
                56 00000000'    00   9E 00009          MOVAB   TOKENDSC, R6                                ;
                5E                10   C2 00010          SUBL2   #16, SP                                     ;
                09          10    A6   E8 00013          BLBS    NETUAF_EXISTS, 1$                           ; 2937
                   00000000G    8F   DD 00017          PUSHL   #UAF$_NAFDNE                                ; 2938
                             00AA   31   DD 0001D          BRW     7$                                          ;
                                56   DD 00020 1$:       PUSHL   R6                                          ; 2943
```

```
                                                00000000'  00  9F 00022          PUSHAB  SD_TOKEN1                                       ;
                                     00000000G  00         02  FB 00028          CALLS   #2, CLI$GET_VALUE                               ;
                                     00000000'  00         01  D0 0002F          MOVL    #1, HEADER_FLAG                                 ; 2945
                                                52         66  3C 00036          MOVZWL  TOKENLEN, R2                                    ; 2950
                                                53     04  A6  D0 00039          MOVL    TOKENPTR, R3                                    ;
                                          63    52         25  3A 0003D          LOCC    #37, R2, (R3)                                   ;
                                                           02  12 00041          BNEQ    2$                                              ;
                                                           51  D4 00043          CLRL    R1                                              ;
                                                           51  D5 00045  2$:     TSTL    R1                                              ;
                                                           0C  12 00047          BNEQ    4$                                              ;
                                          63    52         2A  3A 00049          LOCC    #42, R2, (R3)                                   ; 2951
                                                           02  12 0004D          BNEQ    3$                                              ;
                                                           51  D4 0004F          CLRL    R1                                              ;
                                                           51  D5 00051  3$:     TSTL    R1                                              ;
                                                           0F  13 00053          BEQL    5$                                              ;
                                       F8  A6              01  D0 00055  4$:     MOVL    #1, WILD_NETUSER                                ; 2954
                                       F4  A6              52  D0 00059          MOVL    R2, MATCH_TOKENLEN                              ; 2955
                                 B0    A6        63        52  28 0005D          MOVC3   R2, (R3), MATCH_TOKEN                           ; 2956
                                                           29  11 00062          BRB     6$                                              ; 2950
                                                F8     A6  D4 00064  5$:     CLRL    WILD_NETUSER                                    ; 2963
                                                           5E  DD 00067          PUSHL   SP                                              ; 2964
                                                08     AE  9F 00069          PUSHAB  REMUSER_PTR                                     ;
                                                10     AE  9F 0006C          PUSHAB  NODE_LEN                                        ;
                                                18     AE  9F 0006F          PUSHAB  NODE_PTR                                        ;
                                           F412     CF        04  FB 00072          CALLS   #4, REMOTE_PARSE                                ;
                                                61         50  E9 00077          BLBC    R0, 9$                                          ;
                20                 20        0C    BE     08  AE  2C 0007A          MOVC5   NODE_LEN, @NODE_PTR, #32, #32, NETBUF           ; 2967
                                                     05A4  C6      00081
                20                 20        04    BE         6E  2C 00084          MOVC5   REMUSER_LEN, @REMUSER_PTR, #32, #32, -          ; 2969
                                                     05C4  C6      0008A                  NETBUF+32
                                           98    A6     069C  C6  9E 0008D  6$:     MOVAB   OUTRAB, RABPTR                                  ; 2975
                                                50 00000000V  00  9E 00093          MOVAB   DISPLAY_PROXY, ACTION                           ; 2976
                                                     06E0  C6  D4 0009A          CLRL    FOUND_MATCH                                     ; 2977
                                          0650    C6 00100008  8F  D0 0009E          MOVL    #1048584, NAFRAB+4                              ; 2978
                                                           50  DD 000A7          PUSHL   ACTION                                          ; 2983
                                     00000000V  00         01  FB 000A9          CALLS   #1, LOCATE_PROXY                                ;
                                           18    A6        50  D0 000B0          MOVL    RMSERR, R0                                      ;
                                                24         50  E8 000B4          BLBS    R0, 9$                                          ;
                                                50     18  A6  D0 000B7          MOVL    RMSERR, R0                                      ; 2986
                                     000182B2    8F        50  D1 000BB          CMPL    R0, #98994                                      ;
                                                           0A  12 000C2          BNEQ    8$                                              ;
                                                00000000G  8F  DD 000C4          PUSHL   #UAF$_BADSPC                                    ; 2988
                                                67         01  FB 000CA  7$:     CALLS   #1, LIB$SIGNAL                                  ;
                                                           04  000CD          RET                                                     ;
                                                           50  DD 000CE  8$:     PUSHL   R0                                              ; 2990
                                                           7E  D4 000D0          CLRL    -(SP)                                           ;
                                                00000000G  8F  DD 000D2          PUSHL   #UAF$_SHOW_ERR                                  ;
                                                67         03  FB 000D8          CALLS   #3, LIB$SIGNAL                                  ;
                                                           04  000DB  9$:     RET                                                     ; 2992

; Routine Size:  220 bytes,    Routine Base:  $CODE$ + 11B3
```

```
;  2918     2993  1 %sbttl 'locate_proxy - access given proxy record (s)'
;  2919     2994  1 routine locate_proxy (action) =
;  2920     2995  2 begin
;  2921     2996  2
;  2922     2997  2 !++
;  2923     2998  2 !
;  2924     2999  2 ! FUNCTIONAL DESCRIPTION:
;  2925     3000  2 !
;  2926     3001  2 !       This routine will call a requested action routine a number of times.
;  2927     3002  2 !
;  2928     3003  2 ! INPUTS:
;  2929     3004  2 !
;  2930     3005  2 !       ACTION - the action routine to call for each NEUAF record
;  2931     3006  2 !
;  2932     3007  2 ! OUTPUTS:
;  2933     3008  2 !
;  2934     3009  2 !       none
;  2935     3010  2 !
;  2936     3011  2 ! SIDE EFFECTS:
;  2937     3012  2 !
;  2938     3013  2 !       none
;  2939     3014  2 !
;  2940     3015  2 !--
;  2941     3016  2
;  2942     3017  2 local
;  2943     3018  2         status;
;  2944     3019  2
;  2945     3020  2 !
;  2946     3021  2 ! If wild user, set acces to sequential and fetch all records
;  2947     3022  2 !
;  2948     3023  2 if .wild_netuser
;  2949     3024  2 then
;  2950     3025  3     begin
;  2951     3026  3     nafrab[rab$b_rac] = rab$c_seq;
;  2952     3027  3     $rewind (rab = nafrab);
;  2953     3028  2     end;
;  2954     3029  2
;  2955     3030  2 status = get_proxy_record ();
;  2956     3031  2
;  2957     3032  2 !
;  2958     3033  2 ! Fetch record and call action routine until EOF
;  2959     3034  2 !
;  2960     3035  2 if .status
;  2961     3036  2 then (.action) ();
;  2962     3037  2
;  2963     3038  2 if .wild_netuser
;  2964     3039  2 then
;  2965     3040  3     begin
;  2966     3041  3     while status = get_proxy_record ()
;  2967     3042  3     do (.action) ();
;  2968     3043  3     if .status eql rms$_eof
;  2969     3044  3     then status = true;
;  2970     3045  2     end;
;  2971     3046  2
;  2972     3047  2 !
;  2973     3048  2 ! Restore keyed access
;  2974     3049  2 !
```

```
;  2975          3050  2 nafrab[rab$b_rac] = rab$c_key;
;  2976          3051  2
;  2977          3052  2 if .wild_netuser and not .found_match
;  2978          3053  2 then LIB$SIGNAL(UAF$_BADSPC);
;  2979          3054  2
;  2980          3055  2 .status
;  2981          3056  1 end;


                                        001C 00000 LOCATE_PROXY:
                                                   .WORD   Save R2,R3,R4                    ; 2994
                    54 00000000V    00  9E 00002           MOVAB   GET_PROXY_RECORD, R4
                    53 00000000'    00  9E 00009           MOVAB   WILD_NETUSER, R3
                    0F              63  E9 00010           BLBC    WILD_NETUSER, 1$         ; 3023
                           0672     C3  94 00013           CLRB    NAFRAB+30                ; 3026
                           0654     C3  9F 00017           PUSHAB  NAFRAB                   ; 3027
              00000000G     00            01  FB 0001B           CALLS   #1, SYS$REWIND
                    64              00  FB 00022 1$:       CALLS   #0, GET_PROXY_RECORD     ; 3030
                    52              50  D0 00025           MOVL    R0, STATUS
                    04              52  E9 00028           BLBC    STATUS, 2$               ; 3035
                 04 BC              00  FB 0002B           CALLS   #0, @ACTION              ; 3036
                    1B              63  E9 0002F 2$:       BLBC    WILD_NETUSER, 5$         ; 3038
                    64              00  FB 00032 3$:       CALLS   #0, GET_PROXY_RECORD     ; 3041
                    52              50  D0 00035           MOVL    R0, STATUS
                    06              52  E9 00038           BLBC    STATUS, 4$
                 04 BC              00  FB 0003B           CALLS   #0, @ACTION              ; 3042
                    F1              11  0003F           BRB     3$
              0001827A 8F           52  D1 00041 4$:       CMPL    STATUS, #98938           ; 3043
                                    03  12 00048           BNEQ    5$
                    52              01  D0 0004A           MOVL    #1, STATUS               ; 3044
                 0672 C3            01  90 0004D 5$:       MOVB    #1, NAFRAB+30            ; 3050
                    12              63  E9 00052           BLBC    WILD_NETUSER, 6$         ; 3052
                    0D     06E8     C3  E8 00055           BLBS    FOUND_MATCH, 6$
                       00000000G    8F  DD 0005A           PUSHL   #UAF$_BADSPC             ; 3053
              00000000G     00            01  FB 00060           CALLS   #1, LIB$SIGNAL
                    50              52  D0 00067 6$:       MOVL    STATUS, R0               ; 3056
                                        04 0006A           RET

; Routine Size:  107 bytes,    Routine Base:  $CODE$ + 128F
```

```
;  2983          3057  1 %sbttl 'get_proxy_record - read single proxy record'
;  2984          3058  1 routine get_proxy_record =
;  2985          3059  2 begin
;  2986          3060  2
;  2987          3061  2 !++
;  2988          3062  2 !
;  2989          3063  2 ! FUNCTIONAL DESCRIPTION:
;  2990          3064  2 !
;  2991          3065  2 !       This routine accesses a specific NETUAF.DAT record
;  2992          3066  2 !
;  2993          3067  2 ! INPUTS:
;  2994          3068  2 !
;  2995          3069  2 !       none
;  2996          3070  2 !
;  2997          3071  2 ! OUTPUTS:
;  2998          3072  2 !
;  2999          3073  2 !       none
;  3000          3074  2 !
;  3001          3075  2 ! SIDE EFFECTS:
;  3002          3076  2 !
;  3003          3077  2 !       none
;  3004          3078  2 !
;  3005          3079  2 !--
;  3006          3080  2
;  3007          3081  2 local
;  3008          3082  2         counter,
;  3009          3083  2         success;
;  3010          3084  2
;  3011          3085  2 counter = retry_rlk;
;  3012          3086  2
;  3013          3087  3 while ((success = $get (rab = nafrab)) eql rms$_rlk)
;  3014          3088  3       and  ((counter = .counter - 1) geq 0)
;  3015          3089  2 do
;  3016          3090  2    if $schdwk (daytim = wakedelta) then $hiber;
;  3017          3091  2
;  3018          3092  2 .success
;  3019          3093  1 end;


                                                        .EXTRN  SYS$GET, SYS$SCHDWK
                                                        .EXTRN  SYS$HIBER

                                        000C 00000 GET_PROXY_RECORD:
                                                        .WORD   Save R2,R3                      ;3058
                        52                      08  D0 00002           MOVL    #8, COUNTER     ;3085
                                00000000'       00  9F 00005 1$:       PUSHAB  NAFRAB          ;3087
00000000G               00                      01  FB 0000B           CALLS   #1, SYS$GET
                        53                      50  D0 00012           MOVL    R0, SUCCESS
000182AA                8F                      53  D1 00015           CMPL    SUCCESS, #98986
                                                21  12 0001C           BNEQ    2$
                                                52  D7 0001E           DECL    COUNTER             ;3088
                                                1D  19 00020           BLSS    2$
                                                7E  D4 00022           CLRL    -(SP)               ;3090
                                00000000'       00  9F 00024           PUSHAB  WAKEDELTA
                                                7E  7C 0002A           CLRQ    -(SP)
00000000G               00                      04  FB 0002C           CALLS   #4, SYS$SCHDWK
```

```
                                 CF              50  E9 00033          BLBC    R0, 1$                                  ;
                      00000000G  00              00  FB 00036          CALLS   #0, SYS$HIBER                           ;
                                                 C6  11 0003D          BRB     1$                                      ;
                                 50              53  D0 0003F 2$:      MOVL    SUCCESS, R0                             ; 3093
                                                     04 00042          RET                                             ;

; Routine Size:  67 bytes,    Routine Base:  $CODE$ + 12FA
```

```
;  3021     3094  1 %sbttl 'display_proxy - format and output a proxy entry'
;  3022     3095  1 routine display_proxy : novalue =
;  3023     3096  2 begin
;  3024     3097  2
;  3025     3098  2 !++
;  3026     3099  2 !
;  3027     3100  2 ! FUNCTIONAL DESCRIPTION:
;  3028     3101  2 !
;  3029     3102  2 !       This routine formats and outputs a line of a NETUAF.DAT entry
;  3030     3103  2 !
;  3031     3104  2 ! INPUTS:
;  3032     3105  2 !
;  3033     3106  2 !       none
;  3034     3107  2 !
;  3035     3108  2 ! OUTPUTS:
;  3036     3109  2 !
;  3037     3110  2 !       none
;  3038     3111  2 !
;  3039     3112  2 ! SIDE EFFECTS:
;  3040     3113  2 !
;  3041     3114  2 !       none
;  3042     3115  2 !
;  3043     3116  2 !--
;  3044     3117  2
;  3045     3118  2 bind
;  3046     3119  2         nafhdr          = cstring (' Node   Remote User     Local User'),
;  3047     3120  2         shownaf         = cstring ('!6AD::!12AD    !12AD');
;  3048     3121  2
;  3049     3122  2
;  3050     3123  2 if .wild_netuser
;  3051     3124  2 and not
;  3052     3125  3     begin
;  3053     3126  3     local
;  3054     3127  3         nodelen, usrlen, proxy_buf : vector[naf$s_remname+2,byte];
;  3055     3128  3     map
;  3056     3129  3         dbl_colon : vector;
;  3057     3130  3
;  3058     3131  3     nodelen = namelen (naf$s_node, netbuf[naf$t_node]);
;  3059     3132  3     usrlen  = namelen (naf$s_remuser, netbuf[naf$t_remuser]);
;  3060     3133  3     ch$move (.nodelen, netbuf[naf$t_node], proxy_buf [0]);
;  3061     3134  3     ch$move (2, .dbl_colon[1], proxy_buf [.nodelen]);
;  3062     3135  3     ch$move (.usrlen, netbuf [naf$t_remuser], proxy_buf [.nodelen+2]);
;  3063     3136  3
;  3064     3137  3     usrlen = .nodelen + .usrlen + 2;
;  3065     3138  3     fmg$match_name (.usrlen, proxy_buf, .match_tokenlen, match_token)
;  3066     3139  3     end
;  3067     3140  2 then
;  3068     3141  2     return;
;  3069     3142  2
;  3070     3143  2 found_match = true;
;  3071     3144  2
;  3072     3145  2 if .header_flag
;  3073     3146  2 then
;  3074     3147  3     begin
;  3075     3148  3     faomac (nafhdr);
;  3076     3149  3     output_null;
;  3077     3150  3     header_flag = false;
```

```
;  3078          3151  2     end;
;  3079          3152  2
;  3080        P 3153  2 faomac (shownaf,
;  3081        P 3154  2 !***     naf$s_node,      netbuf[naf$t_node],
;  3082        P 3155  2 !***     naf$s_remuser,   netbuf[naf$t_remuser],
;  3083        P 3156  2 !***     naf$s_localuser, netbuf[naf$t_localuser]);
;  3084        P 3157  2      6,  netbuf[naf$t_node],
;  3085        P 3158  2     12,  netbuf[naf$t_remuser],
;  3086          3159  2     12,  netbuf[naf$t_localuser]);
;  3087          3160  2
;  3088          3161  1 end;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                 22  00254 P.ACN:  .BYTE   34
20  65  74  6F  6D  65  52  20  20  20  65  64  6F  4E  20  00255          .ASCII  \ Node   Remote User     Local User\
20  6C  61  63  6F  4C  20  20  20  20  20  72  65  73  55  00264
                                                 72  65  73  55  00273
                                                 14  00277 P.ACO:  .BYTE   20
20  20  20  20  44  41  32  31  21  3A  3A  44  41  36  21  00278          .ASCII  \!6AD::!12AD    !12AD\
                                                 44  41  32  31  21  00287

                                                NAFHDR=             P.ACN
                                                SHOWNAF=            P.ACO
                                                        .EXTRN  SYS$FAO

                                                        .PSECT  $CODE$,NOWRT,2

                                             0FFC 00000 DISPLAY_PROXY:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11                    ;  3095
                                          5B 0C000000'  00  9E 00002        MOVAB   HEADER_FLAG, R11
                                          5A 00000000G  00  9E 00009        MOVAB   SYS$PUT, R10
                                          59 00000000'  00  9E 00010        MOVAB   DBL_COLON+4, R9
                                          58 00000000'  00  9E 00017        MOVAB   RABPTR, R8
                                          5E       BC   AE  9E 0001E        MOVAB   -68(SP), SP
                                          47       60   A8  E9 00022        BLBC    WILD_NETUSER, 1$                       ;  3123
                                    060C  C8           20      3A 00026     LOCC    #32, #32, NETBUF                       ;  3131
                                          56           20      50 C3 0002C  SUBL3   R0, #32, NODELEN
                                    062C  C8           20      3A 00030     LOCC    #32, #32, NETBUF+32                    ;  3132
                                          57           20      50 C3 00036  SUBL3   R0, #32, USRLEN
                                          6E     060C  C8      56 28 0003A  MOVC3   NODELEN, NETBUF, PROXY_BUF             ;  3133
                                          50           69      D0 00040     MOVL    DBL_COLON+4, R0                        ;  3134
                                                    6E46  9F  00043        PUSHAB  PROXY_BUF[NODELEN]
                                          9E           60      B0 00046     MOVW    (R0), @(SP)+
                                       02 AE46   062C  C8      57 28 00049  MOVC3   USRLEN, NETBUF+32, PROXY_BUF+2[NODELEN]  ;  3135
                                          57       02 A746  9E  00051        MOVAB   2(USRLEN)[NODELEN], USRLEN              ;  3137
                                          55       18   A8  9E  00056        MOVAB   MATCH_TOKEN, R5                        ;  3138
                                          53            6E 9E  0005A        MOVAB   PROXY_BUF, R3
                                          54       5C   A8  D0  0005D        MOVL    MATCH_TOKENLEN, R4
                                          52            57 D0  00061        MOVL    USRLEN, R2
                                             00000000G  00  16  00064        JSB     FMG$MATCH_NAME
                                          6B           50  E9  0006A        BLBC    R0, 3$
                                    0748  C8           01  D0  0006D 1$:     MOVL    #1, FOUND_MATCH                        ;  3143
                                          2F           6B  E9  00072        BLBC    HEADER_FLAG, 2$                        ;  3145
                                       F8 A8      0140   C9  9B  00075        MOVZBW  NAFHDR, FAODSC                         ;  3148
```

```
                          FC    A8      0141    C9  9E 0007B          MOVAB   NAFHDR+1, FAODSC+4                          ;
                                  F0    A8  9F 00081                PUSHAB  DISDSC                                      ;
                    7E                  68      22  C1 00084        ADDL3   #34, RABPTR, -(SP)                          ;
                                  F8    A8  9F 00088                PUSHAB  FAODSC                                      ;
                       00000000G  00          03  FB 0008B        CALLS   #3, SYS$FAO                                 ;
                                              68  DD 00092        PUSHL   RABPTR                                      ;
                                  6A          01  FB 00094        CALLS   #1, SYS$PUT                                 ;
                                  50          68  D0 00097        MOVL    RABPTR, R0                                  ;
                                        22    A0  B4 0009A        CLRW    34(R0)                                      ;
                                              50  DD 0009D        PUSHL   R0                                          ;
                                  6A          01  FB 0009F        CALLS   #1, SYS$PUT                                 ;
                                              6B  D4 000A2        CLRL    HEADER_FLAG                                 ; 3150
                          F8    A8      0163    C9  9B 000A4 2$:    MOVZBW  SHOWNAF, FAODSC                             ; 3159
                          FC    A8      0164    C9  9E 000AA        MOVAB   SHOWNAF+1, FAODSC+4                         ;
                                        064C    C8  9F 000B0        PUSHAB  NETBUF+64                                   ;
                                                0C  DD 000B4        PUSHL   #12                                         ;
                                        062C    C8  9F 000B6        PUSHAB  NETBUF+32                                   ;
                                                0C  DD 000BA        PUSHL   #12                                         ;
                                        060C    C8  9F 000BC        PUSHAB  NETBUF                                      ;
                                                06  DD 000C0        PUSHL   #6                                          ;
                                  F0    A8  9F 000C2                PUSHAB  DISDSC                                      ;
                    7E                  68      22  C1 000C5        ADDL3   #34, RABPTR, -(SP)                          ;
                                  F8    A8  9F 000C9                PUSHAB  FAODSC                                      ;
                       00000000G  00          09  FB 000CC        CALLS   #9, SYS$FAO                                 ;
                                              68  DD 000D3        PUSHL   RABPTR                                      ;
                                  6A          01  FB 000D5        CALLS   #1, SYS$PUT                                 ;
                                              04  000D8 3$:       RET                                                 ; 3161

; Routine Size:  217 bytes,    Routine Base:  $CODE$ + 133D
```

```
;3090    3162  1 %sbttl 'wild_user - user wild card routine'
;3091    3163  1 global routine wild_user (action) =
;3092    3164  2 begin
;3093    3165  2
;3094    3166  2 !++
;3095    3167  2 !
;3096    3168  2 ! FUNCTIONAL DESCRIPTION:
;3097    3169  2 !
;3098    3170  2 !       Provide a general means of accessing the User Authorization File
;3099    3171  2 !       records.  There are six methods:
;3100    3172  2 !
;3101    3173  2 !                               UGMS
;3102    3174  2 !                               IRET
;3103    3175  2 !                               CPMR
;3104    3176  2 !                               \\\\
;3105    3177  2 !                               FWWW
;3106    3178  2 !                               LIII
;3107    3179  2 !                               ALLL
;3108    3180  2 !       Syntax                  GDDD    Interpretation
;3109    3181  2 !       ------                          --------------
;3110    3182  2 !       Username                FFFF    Exactly one user is to be located
;3111    3183  2 !       *                       FFFT    All users 'alphabetically)
;3112    3184  2 !       [Group,Member]          TFFF    All users with the specified UIC
;3113    3185  2 !       [Group,*]               TFTF    All users in the specified group (by member)
;3114    3186  2 !       [*,Member]              TTFF    A FIFO listing of the groups with this member
;3115    3187  2 !       [*,*]                   TTTF    All users by UIC
;3116    3188  2 !
;3117    3189  2 ! INPUTS:
;3118    3190  2 !
;3119    3191  2 !       ACTION - Pointer to routine to call after each successful GET
;3120    3192  2 !
;3121    3193  2 ! IMPLICIT INPUTS:
;3122    3194  2 !
;3123    3195  2 !       UIC_FLAG - UIC form (instead of username)
;3124    3196  2 !       GRP_WILD - Group wild card (must imply UIC_FLAG)
;3125    3197  2 !       MEM_WILD - Member wild card (must imply UIC_FLAG)
;3126    3198  2 !       STR_WILD - all users alphabetically (must imply NOT UIC_FLAG)
;3127    3199  2 !       UAFRAB - RMS data structure for SYSUAF.DAT
;3128    3200  2 !       RECBUF - The current record
;3129    3201  2 !
;3130    3202  2 ! OUTPUTS:
;3131    3203  2 !
;3132    3204  2 !       none
;3133    3205  2 !
;3134    3206  2 ! IMPLICIT OUTPUTS:
;3135    3207  2 !
;3136    3208  2 !       none
;3137    3209  2 !
;3138    3210  2 ! ROUTINE VALUE:
;3139    3211  2 !
;3140    3212  2 !       If an abnormal condition is encountered the appropriate status
;3141    3213  2 !       is returned.
;3142    3214  2 !
;3143    3215  2 ! SIDE EFFECTS:
;3144    3216  2 !
;3145    3217  2 !       none
;3146    3218  2 !--
```

```
;  3147     3219  2
;  3148     3220  2 macro
;  3149     3221  2         lmt_l_uic = 0,0,32,0%,          ! User ID Code
;  3150     3222  2         lmt_w_mem = 0,0,16,0%,          ! Member subfield
;  3151     3223  2         lmt_w_grp = 2,0,16,0%;          ! Group subfield
;  3152     3224  2
;  3153     3225  2 local
;  3154     3226  2         status,                         ! This routine's status
;  3155     3227  2         lmtkey  : block[4,byte];        ! Limiting key value for sequential loop
;  3156     3228  2
;  3157     3229  2 if .uic_flag
;  3158     3230  2 then
;  3159     3231  2 !
;  3160     3232  2 ! Change the key of reference and the key buffer if a UIC form was
;  3161     3233  2 ! specified.
;  3162     3234  2 !
;  3163     3235  3     begin
;  3164     3236  3     uafrab[rab$b_krf] = 1;
;  3165     3237  3     uafrab[rab$l_kbf] = recbuf[uaf$l_uic];
;  3166     3238  3     uafrab[rab$b_ksz] = 4;
;  3167     3239  2     end;
;  3168     3240  2
;  3169     3241  2 if .mem_wild and not .grp_wild
;  3170     3242  2 then
;  3171     3243  2 !
;  3172     3244  2 ! The UIC requested was of the form [Group,*]
;  3173     3245  2 !
;  3174     3246  2     uafrab[rab$v_kge] = true;
;  3175     3247  2
;  3176     3248  2 !
;  3177     3249  2 ! LMTKEY need be loaded only IF .UIC_FLAG AND NOT (.GRP_WILD AND .MEM_WILD)
;  3178     3250  2 ! but it is simpler to always load it.
;  3179     3251  2 !
;  3180     3252  2
;  3181     3253  2 lmtkey[lmt_l_uic] = .recbuf[uaf$l_uic];
;  3182     3254  2
;  3183     3255  2 !
;  3184     3256  2 ! Locate the first user meeting the specification.
;  3185     3257  2 !
;  3186     3258  2
;  3187     3259  2 if .str_wild or .grp_wild
;  3188     3260  2 then
;  3189     3261  3     begin
;  3190     3262  3 !
;  3191     3263  3 ! Every user in the file is to be accessed.
;  3192     3264  3 !
;  3193     3265  3     uafrab[rab$b_rac] = rab$c_seq;
;  3194     3266  3     $rewind (rab = uafrab);
;  3195     3267  3     status = get_uaf_record ();
;  3196     3268  3     end
;  3197     3269  2 else
;  3198     3270  3     begin
;  3199     3271  3     status = get_uaf_record ();
;  3200     3272  3     if .uic_flag
;  3201     3273  3     then
;  3202     3274  4         begin
;  3203     3275  4 !
```

```
; 3204     3276  4 ! Even an explicit UIC requires sequential reads to locate duplicates.
; 3205     3277  4 !
; 3206     3278  4           uafrab[rab$b_rac] = rab$c_seq;
; 3207     3279  4           if .mem_wild and not .grp_wild
; 3208     3280  4           then
; 3209     3281  5               begin
; 3210     3282  5 !
; 3211     3283  5 ! RAB$V_KGE is set on the initial access for specifications of the
; 3212     3284  5 ! form [Group,*] so if the specified group has no members the record
; 3213     3285  5 ! will be that of a user in another group.
; 3214     3286  5 !
; 3215     3287  5               uafrab[rab$v_kge] = false;
; 3216     3288  5               if .lmtkey[lmt_w_grp] nequ .recbuf[uaf$w_grp]
; 3217     3289  5               then
; 3218     3290  5                   status = rms$_rnf;
; 3219     3291  4               end;
; 3220     3292  3           end;
; 3221     3293  2       end;
; 3222     3294  2
; 3223     3295  2 if .status
; 3224     3296  2 then
; 3225     3297  3     begin
; 3226     3298  3 !
; 3227     3299  3 ! Feed the action routine the first record.  In the case of an explicit
; 3228     3300  3 ! username specification this will be the only record.
; 3229     3301  3 !
; 3230     3302  3     while .status
; 3231     3303  3     do
; 3232     3304  4         begin
; 3233     3305  5         if (
; 3234     3306  5             if .grp_wild and not .mem_wild
; 3235     3307  5             then .recbuf[uaf$w_mem] eql .lmtkey[lmt_w_mem]
; 3236     3308  5             else true
; 3237     3309  5 )
; 3238     3310  4         then status = (.action) ();
; 3239     3311  4         if not .status then exitloop;
; 3240     3312  5         if not (.str_wild or .uic_flag)
; 3241     3313  4         then exitloop;
; 3242     3314  4         status = get_uaf_record ();
; 3243     3315  4         if not .status then exitloop;
; 3244     3316  4         if .uic_flag
; 3245     3317  4         then
; 3246     3318  5             begin
; 3247     3319  5 !
; 3248     3320  5 ! The limiting key value is used in different ways depending
; 3249     3321  5 ! on the form of the UIC specification.
; 3250     3322  5 !
; 3251     3323  5             if .mem_wild and not .grp_wild
; 3252     3324  5             then
; 3253     3325  5 !
; 3254     3326  5 ! [Group,*]
; 3255     3327  5 !
; 3256     3328  6                 begin
; 3257     3329  6                 if .lmtkey[lmt_w_grp] nequ .recbuf[uaf$w_grp]
; 3258     3330  6                 then exitloop;
; 3259     3331  5                 end;
; 3260     3332  6             if not (.grp_wild or .mem_wild)
```

```
;  3261          3333  5              then
;  3262          3334  5 !
;  3263          3335  5 ! [Group,Member]
;  3264          3336  5 !
;  3265          3337  6                  begin
;  3266          3338  6                  if .lmtkey[lmt_l_uic] nequ .recbuf[uaf$l_uic]
;  3267          3339  6                  then exitloop;
;  3268          3340  5                  end;
;  3269          3341  4              end;
;  3270          3342  3          end;                                ! end of wild carding loop
;  3271          3343  3      if .status eql rms$_eof
;  3272          3344  3      then status = true;                     ! Hitting EOF is ok
;  3273          3345  2      end;
;  3274          3346  2
;  3275          3347  2 if .str_wild and not .found_match
;  3276          3348  2 then LIB$SIGNAL(UAF$_BADSPC);
;  3277          3349  2
;  3278          3350  2 !
;  3279          3351  2 ! The RAB must be returned to its former state before exiting.
;  3280          3352  2 !
;  3281          3353  2 uafrab[rab$b_rac] = rab$c_key;
;  3282          3354  2
;  3283          3355  2 if .uic_flag
;  3284          3356  2 then
;  3285          3357  3      begin
;  3286          3358  3      uafrab[rab$b_krf] = 0;
;  3287          3359  3      uafrab[rab$l_kbf] = recbuf[uaf$t_username];
;  3288          3360  3      uafrab[rab$b_ksz] = uaf$s_username;
;  3289          3361  2      end;
;  3290          3362  2
;  3291          3363  2 !
;  3292          3364  2 ! Reset parse count
;  3293          3365  2 !
;  3294          3366  2 call_count = 0;
;  3295          3367  2
;  3296          3368  2 .status
;  3297          3369  1 end;


                                                001C 00000          .ENTRY  WILD_USER, Save R2,R3,R4                        ; 3163
                                    54 00000000V    00    9E 00002          MOVAB   GET_UAF_RECORD, R4                      ;
                                    53 00000000'    00    9E 00009          MOVAB   GRP_WILD, R3                            ;
                                    5E                    04    C2 00010    SUBL2   #4, SP                                  ;
                                    0E                    FC    A3    E9 00013    BLBC    UIC_FLAG, 1$                      ; 3229
                                FF50    C3          F95C    C3    9E 00017          MOVAB   RECBUF+36, UAFRAB+48            ; 3237
                                FF54    C3          0104    8F    B0 0001E          MOVW    #260, UAFRAB+52                 ; 3238
                                    08                    04    A3    E9 00025 1$:    BLBC    MEM_WILD, 2$              ; 3241
                                    05                          63    E8 00029    BLBS    GRP_WILD, 2$                    ;
                                FF26    C3                      20    88 0002C    BISB2   #32, UAFRAB+6                   ; 3246
                                    6E          F95C    C3    D0 00031 2$:    MOVL    RECBUF+36, LMTKEY                   ; 3253
                                    03                    08    A3    E8 00036    BLBS    STR_WILD, 3$                      ; 3259
                                    17                          63    E9 0003A    BLBC    GRP_WILD, 4$                    ;
                                                FF3E    C3    94 0003D 3$:    CLRB    UAFRAB+30                           ; 3265
                                                FF20    C3    9F 00041          PUSHAB  UAFRAB                          ; 3266
```

```
                00000000G  00              01  FB 00045          CALLS   #1, SYS$REWIND                          ;
                           64              00  FB 0004C          CALLS   #0, GET_UAF_RECORD                     ; 3267
                           52              50  D0 0004F          MOVL    R0, STATUS
                                           29  11 00052          BRB     5$                                     ; 3259
                           64              00  FB 00054 4$:      CALLS   #0, GET_UAF_RECORD                     ; 3271
                           52              50  D0 00057          MOVL    R0, STATUS
                           1F        FC    A3  E9 0005A          BLBC    UIC_FLAG, 5$                           ; 3272
                                 FF3E      C3  94 0005E          CLRB    UAFRAB+30                              ; 3278
                           17        04    A3  E9 00062          BLBC    MEM_WILD, 5$                           ; 3279
                           14              63  E8 00066          BLBS    GRP_WILD, 5$
                     FF26  C3              20  8A 00069          BICB2   #32, UAFRAB+6                          ; 3287
                     F95E  C3        02    AE  B1 0006E          CMPW    LMTKEY+2, RECBUF+38                    ; 3288
                                           07  13 00074          BEQL    5$
                           52  000182B2    8F  D0 00076          MOVL    #98994, STATUS                         ; 3290
                           5B              52  E9 0007D 5$:      BLBC    STATUS, 12$                            ; 3295
                           4C              52  E9 00080 6$:      BLBC    STATUS, 11$                            ; 3302
                           0B              63  E9 00083          BLBC    GRP_WILD, 7$                           ; 3306
                           07        04    A3  E8 00086          BLBS    MEM_WILD, 7$
                           6E      F95C    C3  B1 0008A          CMPW    RECBUF+36, LMTKEY                      ; 3307
                                           07  12 0008F          BNEQ    8$
                     04    BC              00  FB 00091 7$:      CALLS   #0, @ACTION                            ; 3310
                           52              50  D0 00095          MOVL    R0, STATUS
                           34              52  E9 00098 8$:      BLBC    STATUS, 11$                            ; 3311
                           04        08    A3  E8 0009B          BLBS    STR_WILD, 9$                           ; 3312
                           2C        FC    A3  E9 0009F          BLBC    UIC_FLAG, 11$
                           64              00  FB 000A3 9$:      CALLS   #0, GET_UAF_RECORD                     ; 3314
                           52              50  D0 000A6          MOVL    R0, STATUS
                           23              52  E9 000A9          BLBC    STATUS, 11$                            ; 3315
                           D0        FC    A3  E9 000AC          BLBC    UIC_FLAG, 6$                           ; 3316
                           50        04    A3  D0 000B0          MOVL    MEM_WILD, R0                           ; 3323
                           0B              50  E9 000B4          BLBC    R0, 10$
                           C6              63  E8 000B7          BLBS    GRP_WILD, 6$
                     F95E  C3        02    AE  B1 000BA          CMPW    LMTKEY+2, RECBUF+38                    ; 3329
                                           0D  12 000C0          BNEQ    11$
                           BB              63  E8 000C2 10$:     BLBS    GRP_WILD, 6$                           ; 3332
                           B8              50  E8 000C5          BLBS    R0, 6$
                     F95C  C3              6E  D1 000C8          CMPL    LMTKEY, RECBUF+36                      ; 3338
                                           B1  13 000CD          BEQL    6$
                0001827A   8F              52  D1 000CF 11$:     CMPL    STATUS, #98938                         ; 3343
                                           03  12 000D6          BNEQ    12$
                           52              01  D0 000D8          MOVL    #1, STATUS                             ; 3344
                           11        08    A3  E9 000DB 12$:     BLBC    STR_WILD, 13$                          ; 3347
                           0D        F8    A3  E8 000DF          BLBS    FOUND_MATCH, 13$
                                 00000000G   8F  DD 000E3          PUSHL   #UAF$_BADSPC                           ; 3348
                00000000G  00              01  FB 000E9          CALLS   #1, LIB$SIGNAL
                     FF3E  C3              01  90 000F0 13$:     MOVB    #1, UAFRAB+30                          ; 3353
                           0C        FC    A3  E9 000F5          BLBC    UIC_FLAG, 14$                          ; 3355
                     FF50  C3      F93C    C3  9E 000F9          MOVAB   RECBUF+4, UAFRAB+48                    ; 3359
                     FF54  C3              20  B0 00100          MOVW    #32, UAFRAB+52                         ; 3360
                                     F914    C3  D4 00105 14$:     CLRL    CALL_COUNT                             ; 3366
                           50                52  D0 00109          MOVL    STATUS, R0                             ; 3369
                                             04  0010C          RET                                            ;

; Routine Size:  269 bytes,    Routine Base:  $CODE$ + 1416
```

```
;  3299     3370  1 %sbttl 'display_brief - writes a brief user display'
;  3300     3371  1 routine display_brief =
;  3301     3372  2 begin
;  3302     3373  2
;  3303     3374  2 !++
;  3304     3375  2 !
;  3305     3376  2 ! FUNCTIONAL DESCRIPTION:
;  3306     3377  2 !
;  3307     3378  2 !       Provide an ASCII listing of the most important record information
;  3308     3379  2 !       (username, owner, etc.) for each record supplied.
;  3309     3380  2 !
;  3310     3381  2 ! INPUTS:
;  3311     3382  2 !
;  3312     3383  2 !       none
;  3313     3384  2 !
;  3314     3385  2 ! IMPLICIT INPUTS:
;  3315     3386  2 !
;  3316     3387  2 !       RABPTR - RMS data structure for the file
;  3317     3388  2 !
;  3318     3389  2 ! OUTPUTS:
;  3319     3390  2 !
;  3320     3391  2 !       none
;  3321     3392  2 !
;  3322     3393  2 ! IMPLICIT OUTPUTS:
;  3323     3394  2 !
;  3324     3395  2 !       none
;  3325     3396  2 !
;  3326     3397  2 ! ROUTINE VALUE:
;  3327     3398  2 !
;  3328     3399  2 !       none
;  3329     3400  2 !
;  3330     3401  2 ! SIDE EFFECTS:
;  3331     3402  2 !
;  3332     3403  2 !       none
;  3333     3404  2 !--
;  3334     3405  2
;  3335     3406  2 bind
;  3336   P 3407  2 lststr1 = cstring ('       Owner         Username            UIC       Account  Privs',
;  3337     3408  2                   ' Pri Directory'),
;  3338     3409  2 lststr2 = cstring ('!20AC !12AD !15%U !8AF !6AC !2UL !AC!AC');
;  3339     3410  2
;  3340     3411  2 !
;  3341     3412  2 ! Output a header if one was requested.
;  3342     3413  2 !
;  3343     3414  2 if .header_flag
;  3344     3415  2 then
;  3345     3416  3     begin
;  3346     3417  3     faomac (lststr1);
;  3347     3418  3     output_null;
;  3348     3419  3     header_flag = false;
;  3349     3420  2     end;
;  3350     3421  2
;  3351     3422  2 if .str_wild and not fmg$match_name (namelen (uaf$s_username,recbuf[uaf$t_username]),
;  3352     3423  2                                       recbuf[uaf$t_username],
;  3353     3424  2                                       .match_token[en,  match_token)
;  3354     3425  2 then return true;
;  3355     3426  2
```

```
;  3356          3427  2 found_match = true;
;  3357          3428  2
;  3358          3429  2 !
;  3359          3430  2 ! Output the record.
;  3360          3431  2 !
;  3361          3432  2
;  3362          3433  2 ch$fill (' ', disbuflen, disbuf);
;  3363          3434  2
;  3364        P 3435  2 faomac (lststr2,
;  3365        P 3436  2     recbuf[uaf$t_owner],
;  3366        P 3437  2 !***     uaf$s_username, recbuf[uaf$t_username],
;  3367        P 3438  2     12, recbuf[uaf$t_username],
;  3368        P 3439  2     .recbuf[uaf$l_uic],
;  3369        P 3440  2 !***     uaf$s_account, recbuf[uaf$t_account],
;  3370        P 3441  2     8, recbuf[uaf$t_account],
;  3371        P 3442  2     classify_priv (recbuf[uaf$q_priv], .recbuf[uaf$l_uic]),
;  3372        P 3443  2     .recbuf[uaf$b_pri],
;  3373        P 3444  2     recbuf[uaf$t_defdev],
;  3374          3445  2     recbuf[uaf$t_defdir]);
;  3375          3446  2
;  3376          3447  2 true
;  3377          3448  1 end;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                4E  0028C P.ACP:  .BYTE   78                                    ;
20 20 20 72 65 6E 77 4F 20 20 20 20 20 20 20 20 0028D           .ASCII  \       Owner           Username          \ ;
20 65 6D 61 6E 72 65 73 55 20 20 20 20 20 20 20 0029C                                                           ;
                  20 20 20 20 20 20 20 20 20 20 002AB                                                           ;
75 6F 63 63 41 20 20 20 20 20 20 20 43 49 55 20 002B5           .ASCII  \UIC       Account  Privs Pri Directory\ ;
44 20 69 72 50 20 73 76 69 72 50 20 20 74 6E 20 002C4                                                           ;
                     79 72 6F 74 63 65 72 69 6F 002D3                                                           ;
                                                27  002DB P.ACQ:  .BYTE   39                                    ;
35 31 21 20 44 41 32 31 21 20 43 41 30 32 21 20 002DC           .ASCII  \!20AC !12AD !15%U !8AF !6AC !2UL !AC!AC\ ;
32 21 20 43 41 36 21 20 46 41 38 21 20 55 25 20 002EB                                                           ;
                     43 41 21 43 41 21 20 4C 55 002FA                                                           ;

                                                LSTSTR1=            P.ACP
                                                LSTSTR2=            P.ACQ


                                                        .PSECT  $CODE$,NOWRT,2

                                                07FC 00000 DISPLAY_BRIEF:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10                ; 3371
                                             5A 00000000'   00  9E 00002    MOVAB   HEADER_FLAG, R10                ;
                                             59 00000000G   00  9E 00009    MOVAB   SYS$FAO, R9                     ;
                                             58 00000000G   00  9E 00010    MOVAB   SYS$PUT, R8                     ;
                                             57 00000000'   00  9E 00017    MOVAB   LSTSTR1, R7                     ;
                                             56 00000000'   00  9E 0001E    MOVAB   RABPTR, R6                      ;
                                             28             6A  E9 00025    BLBC    HEADER_FLAG, 1$                 ; 3414
                                          F8 A6             67  9B 00028    MOVZBW  LSTSTR1, FAODSC                 ; 3417
                                          FC A6         01  A7  9E 0002C    MOVAB   LSTSTR1+1, FAODSC+4             ;
                                                    F0  A6  9F 00031    PUSHAB  DISDSC                          ;
                                       7E           66      22  C1 00034    ADDL3   #34, RABPTR, -(SP)              ;
```

```
                                                F8   A6  9F 00038          PUSHAB  FAODSC
                                           69        03  FB 0003B          CALLS   #3, SYS$FAO
                                                     66  DD 0003E          PUSHL   RABPTR
                                           68        01  FB 00040          CALLS   #1, SYS$PUT
                                           50        66  D0 00043          MOVL    RABPTR, R0
                                                   22  A0  B4 00046          CLRW    34(R0)
                                                     50  DD 00049          PUSHL   R0
                                           68        01  FB 0004B          CALLS   #1, SYS$PUT
                                                     6A  D4 0004E          CLRL    HEADER_FLAG                                     3419
                                           23      0758  C6  E9 00050 1$:     BLBC    STR_WILD, 2$                                    3422
                                           55        18  A6  9E 00055          MOVAB   MATCH_TOKEN, R5                                 3423
                                           53      008C  C6  9E 00059          MOVAB   RECBUF+4, R3
                                  008C  C6         20        20  3A 0005E          LOCC    #32, #32, RECBUF+4                      3422
                                           52        E0  A0  9E 00064          MOVAB   -32(R0), R2
                                           52            52  CE 00068          MNEGL   R2, R2
                                           54        5C  A6  D0 0006B          MOVL    MATCH_TOKENLEN, R4                      3423
                                                00000000G  00  16 0006F          JSB     FMG$MATCH_NAME
                                           5D            50  E9 00075          BLBC    R0, 3$
                                    0748   C6                01  D0 00078 2$:     MOVL    #1, FOUND_MATCH                         3427
                0084  8F          20        6E              00  2C 0007D          MOVC5   #0, (SP), #32, #132, DISBUF             3433
                                                 FF6C  C6          00084
                                      F8   A6        4F  A7  9B 00087          MOVZBW  LSTSTR2, FAODSC                         3445
                                      FC   A6        50  A7  9E 0008C          MOVAB   LSTSTR2+1, FAODSC+4
                                                   011C  C6  9F 00091          PUSHAB  RECBUF+148
                                                   00FC  C6  9F 00095          PUSHAB  RECBUF+116
                                           7E      028C  C6  9A 00099          MOVZBL  RECBUF+516, -(SP)
                                                   00AC  C6  DD 0009E          PUSHL   RECBUF+36
                                                   0224  C6  9F 000A2          PUSHAB  RECBUF+412
                                  00000000V  00              02  FB 000A6          CALLS   #2, CLASSIFY_PRIV
                                                     50  DD 000AD          PUSHL   R0
                                                   00BC  C6  9F 000AF          PUSHAB  RECBUF+52
                                                     08  DD 000B3          PUSHL   #8
                                                   00AC  C6  DD 000B5          PUSHL   RECBUF+36
                                                   008C  C6  9F 000B9          PUSHAB  RECBUF+4
                                                     0C  DD 000BD          PUSHL   #12
                                                   00DC  C6  9F 000BF          PUSHAB  RECBUF+84
                                                     F0  A6  9F 000C3          PUSHAB  DISDSC
                                  7E                 66        22  C1 000C6          ADDL3   #34, RABPTR, -(SP)
                                                     F8  A6  9F 000CA          PUSHAB  FAODSC
                                           69        0D  FB 000CD          CALLS   #13, SYS$FAO
                                                     66  DD 000D0          PUSHL   RABPTR
                                           68        01  FB 000D2          CALLS   #1, SYS$PUT
                                           50        01  D0 000D5 3$:      MOVL    #1, R0                                  3448
                                                     04  000D8          RET

; Routine Size:  217 bytes,    Routine Base:  $CODE$ + 1523
```

```
;  3379          3449  1 %sbttl 'classify_priv - classifies contents of priv vector'
;  3380          3450  1 routine classify_priv (prvadr, uic) =
;  3381          3451  2 begin
;  3382          3452  2
;  3383          3453  2 !++
;  3384          3454  2 !
;  3385          3455  2 ! FUNCTIONAL DESCRIPTION:
;  3386          3456  2 !
;  3387          3457  2 !       Classifies privilege bits and reports the highest class available
;  3388          3458  2 !       to the owner of the supplied vector.
;  3389          3459  2 !
;  3390          3460  2 ! INPUTS:
;  3391          3461  2 !
;  3392          3462  2 !       PRVADR - Address of the privilege vector
;  3393          3463  2 !
;  3394          3464  2 ! IMPLICIT INPUTS:
;  3395          3465  2 !
;  3396          3466  2 !       none
;  3397          3467  2 !
;  3398          3468  2 ! OUTPUTS:
;  3399          3469  2 !
;  3400          3470  2 !       none
;  3401          3471  2 !
;  3402          3472  2 ! IMPLICIT OUTPUTS:
;  3403          3473  2 !
;  3404          3474  2 !       none
;  3405          3475  2 !
;  3406          3476  2 ! ROUTINE VALUE:
;  3407          3477  2 !
;  3408          3478  2 !       none
;  3409          3479  2 !
;  3410          3480  2 ! SIDE EFFECTS:
;  3411          3481  2 !
;  3412          3482  2 !       none
;  3413          3483  2 !--
;  3414          3484  2
;  3415          3485  2 map
;  3416          3486  2         prvadr  : ref block[8,byte];
;  3417          3487  2
;  3418          3488  2 bind
;  3419          3489  2         lstprva = cstring ('All'),
;  3420          3490  2         lstprvb = cstring ('Files'),
;  3421          3491  2         lstprvc = cstring ('System'),
;  3422          3492  2         lstprvd = cstring ('Devour'),
;  3423          3493  2         lstprve = cstring ('Group'),
;  3424          3494  2         lstprvf = cstring ('Normal'),
;  3425          3495  2         lstprvg = cstring ('None');
;  3426          3496  2
;  3427          3497  2 if .prvadr[prv$v_cmkrnl]
;  3428          3498  2 or .prvadr[prv$v_cmexec]
;  3429          3499  2 or .prvadr[prv$v_sysnam]
;  3430          3500  2 or .prvadr[prv$v_detach]
;  3431          3501  2 or .prvadr[prv$v_log_io]
;  3432          3502  2 or .prvadr[prv$v_setprv]
;  3433          3503  2 or .prvadr[prv$v_phy_io]
;  3434          3504  2 or .prvadr[prv$v_pfnmap]
;  3435          3505  2 or .prvadr[prv$v_sysprv]
```

```
;  3436          3506  2 or .prvadr[prv$v_readall]
;  3437          3507  2 or .prvadr[prv$v_bypass]
;  3438          3508  3 or (.uic<16, 16> lequ .EXE$GL_SYSUIC)
;  3439          3509  2 then return lstprva;                    ! Universal Privilege
;  3440          3510  2
;  3441          3511  2 if .prvadr[prv$v_diagnose]
;  3442          3512  2 or .prvadr[prv$v_volpro]
;  3443          3513  2 or .prvadr[prv$v_upgrade]
;  3444          3514  2 or .prvadr[prv$v_downgrade]
;  3445          3515  2 or .prvadr[prv$v_security]
;  3446          3516  2 or .prvadr[prv$v_sysgbl]
;  3447          3517  2 then return lstprvb;                    ! Potentially Comprimises File Security
;  3448          3518  2
;  3449          3519  2 if .prvadr[prv$v_pswapm]
;  3450          3520  2 or .prvadr[prv$v_setpri]
;  3451          3521  2 or .prvadr[prv$v_world]
;  3452          3522  2 or .prvadr[prv$v_oper]
;  3453          3523  2 then return lstprvc;                    ! Can Interfere with System Operation
;  3454          3524  2
;  3455          3525  2 if .prvadr[prv$v_grpnam]
;  3456          3526  2 or .prvadr[prv$v_allspool]
;  3457          3527  2 or .prvadr[prv$v_noacnt]
;  3458          3528  2 or .prvadr[prv$v_prmceb]
;  3459          3529  2 or .prvadr[prv$v_prmmbx]
;  3460          3530  2 or .prvadr[prv$v_exquota]
;  3461          3531  2 or .prvadr[prv$v_bugchk]
;  3462          3532  2 or .prvadr[prv$v_prmgbl]
;  3463          3533  2 or .prvadr[prv$v_shmem]
;  3464          3534  2 then return lstprvd;                    ! Can Devour System Resources
;  3465          3535  2
;  3466          3536  2 if .prvadr[prv$v_group]
;  3467          3537  2 or .prvadr[prv$v_grpprv]
;  3468          3538  2 then return lstprve;                    ! Can Interfere with Group Members
;  3469          3539  2
;  3470          3540  2 if .prvadr[prv$v_tmpmbx]
;  3471          3541  2 or .prvadr[prv$v_netmbx]
;  3472          3542  2 or .prvadr[prv$v_mount]
;  3473          3543  2 then return lstprvf;                    ! Normal Privileges
;  3474          3544  2
;  3475          3545  2 lstprvg                                 ! Not Privileged
;  3476          3546  1 end;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                00000 03  00303 P.ACR:  .BYTE   3
                                                00001 6C  6C  41  00304         .ASCII  \All\
                                                00005 05  00307 P.ACS:  .BYTE   5
                                                00006 73  65  6C  69  46  00308         .ASCII  \Files\
                                                0000C 06  0030D P.ACT:  .BYTE   6
                                                0000D 6D  65  74  73  79  53  0030E         .ASCII  \System\
                                                00014 06  00314 P.ACU:  .BYTE   6
                                                00015 72  75  6F  76  65  44  00315         .ASCII  \Devour\
                                                0001C 05  0031B P.ACV:  .BYTE   5
                                                0001D 70  75  6F  72  47  0031C         .ASCII  \Group\
                                                00023 06  00321 P.ACW:  .BYTE   6
```

```
                                                     ;*****************************************************************************
                                                     ;
                                                     ; classify_priv - classifies contents of priv vec
                                                     ;
                                                     ; Inputs:
                                                     ;
                                                     ;       none
                                                     ;
                                                     ; Outputs:
                                                     ;
                                                     ;       R0      address of classification string
                                                     ;
                                                     ;*****************************************************************************

                                                     P.ACX:  .ASCII  \Normal\                                        ;
                                                             .BYTE   4                                               ;
                                                             .ASCII  \None\                                          ;

                                                     LSTPRVA=        P.ACR
                                                     LSTPRVB=        P.ACS
                                                     LSTPRVC=        P.ACT
                                                     LSTPRVD=        P.ACU
                                                     LSTPRVE=        P.ACV
                                                     LSTPRVF=        P.ACW
                                                     LSTPRVG=        P.ACX


                                                             .PSECT  $CODE$,NOWRT,2

                                                     CLASSIFY_PRIV:
                                                             .WORD   Save R2                                         ; 3450
                                  52 00000000'   00  9E      MOVAB   LSTPRVA, R2                                     ;
                                  50       04    AC  D0      MOVL    PRVADR, R0                                      ; 3497
                                  35             60  E8      BLBS    (R0), 1$                                        ;
                          31      60             01  E0      BBS     #1, (R0), 1$                                    ; 3498
                          2D      60             02  E0      BBS     #2, (R0), 1$                                    ; 3499
                          29      60             05  E0      BBS     #5, (R0), 1$                                    ; 3500
                                  60             60  95      TSTB    (R0)                                            ; 3501
                                                 25  19      BLSS    1$                                              ;
                          21      60             0E  E0      BBS     #14, (R0), 1$                                   ; 3502
                          1D      60             16  E0      BBS     #22, (R0), 1$                                   ; 3503
                          19      60             1A  E0      BBS     #26, (R0), 1$                                   ; 3504
                          15      60             1C  E0      BBS     #28, (R0), 1$                                   ; 3505
                          10  04  A0             03  E0      BBS     #3, 4(R0), 1$                                   ; 3506
                          0C      60             1D  E0      BBS     #29, (R0), 1$                                   ; 3507
00000000G  00     0A    AC      10             00  ED      CMPZV   #0, #16, UIC+2, EXE$GL_SYSUIC                   ; 3508
                                                 05  1A      BGTRU   2$                                              ;
                                  51             62  9E  1$: MOVAB   LSTPRVA, R1                                    ; 3509
                                                 6D  11      BRB     10$                                             ;
                          16      60             06  E0  2$: BBS     #6, (R0), 3$                                    ; 3511
                          12      60             15  E0      BBS     #21, (R0), 3$                                   ; 3512
                                  0E         04  A0  E8      BLBS    4(R0), 3$                                       ; 3513
                          09  04  A0             01  E0      BBS     #1, 4(R0), 3$                                   ; 3514
                          04  04  A0             06  E0      BBS     #6, 4(R0), 3$                                   ; 3515
                          06      60             19  E1      BBC     #25, (R0), 4$                                   ; 3516
                                  51         04  A2  9E  3$: MOVAB   LSTPRVB, R1                                    ; 3517
                                                 4D  11      BRB     10$                                             ;
                          0C      60             0C  E0  4$: BBS     #12, (R0), 5$                                   ; 3519
                          08      60             0D  E0      BBS     #13, (R0), 5$                                   ; 3520
                                  04         02  A0  E8      BLBS    2(R0), 5$                                       ; 3521
                          06      60             12  E1      BBC     #18, (R0), 6$                                   ; 3522
                                  51         0A  A2  9E  5$: MOVAB   LSTPRVC, R1                                    ; 3523
                                                 37  11      BRB     10$                                             ;
                          20      60             03  E0  6$: BBS     #3, (R0), 7$                                    ; 3525
                          1C      60             04  E0      BBS     #4, (R0), 7$                                    ; 3526
                          18      60             09  E0      BBS     #9, (R0), 7$                                    ; 3527
                          14      60             0A  E0      BBS     #10, (R0), 7$                                   ; 3528
                          10      60             0B  E0      BBS     #11, (R0), 7$                                   ; 3529
                          0C      60             13  E0      BBS     #19, (R0), 7$                                   ; 3530
                          08      60             17  E0      BBS     #23, (R0), 7$                                   ; 3531
                                  04         03  A0  E8      BLBS    3(R0), 7$                                       ; 3532
```

```
                06              60              1B  E1 000A0            BBC     #27, (R0), 8$                           ; 3533
                                51          11  A2  9E 000A4 7$:       MOVAB   LSTPRVD, R1                             ; 3534
                                                0D  11 000A8            BRB     10$                                     ;
                                05          01  A0  E8 000AA 8$:       BLBS    1(R0), 9$                               ; 3536
                08      04      A0              02  E1 000AE            BBC     #2, 4(R0), 11$                          ; 3537
                                51          18  A2  9E 000B3 9$:       MOVAB   LSTPRVE, R1                             ; 3538
                                50              51  D0 000B7 10$:      MOVL    R1, R0                                  ;
                                                04 000BA                RET                                             ;
                                                60  B5 000BB 11$:      TSTW    (R0)                                    ; 3540
                                                08  19 000BD           BLSS    12$                                     ;
                04              60              14  E0 000BF           BBS     #20, (R0), 12$                          ; 3541
                05              60              11  E1 000C3           BBC     #17, (R0), 13$                          ; 3542
                                50          1E  A2  9E 000C7 12$:      MOVAB   LSTPRVF, R0                             ; 3543
                                                04 000CB                RET                                             ;
                                50          25  A2  9E 000CC 13$:      MOVAB   LSTPRVG, R0                             ; 3451
                                                04 000D0                RET                                             ; 3546

; Routine Size:  209 bytes,    Routine Base:  $CODE$ + 15FC
```

```
;3478     3547  1 %sbttl 'display_full - writes the full user display'
;3479     3548  1 routine display_full =
;3480     3549  2 begin
;3481     3550  2
;3482     3551  2 !++
;3483     3552  2 !
;3484     3553  2 ! FUNCTIONAL DESCRIPTION:
;3485     3554  2 !
;3486     3555  2 !       Display the fields of a UAF record.
;3487     3556  2 !
;3488     3557  2 ! INPUTS:
;3489     3558  2 !
;3490     3559  2 !       RABPTR - RMS data structure for the file
;3491     3560  2 !
;3492     3561  2 ! IMPLICIT INPUTS:
;3493     3562  2 !
;3494     3563  2 !       none
;3495     3564  2 !
;3496     3565  2 ! OUTPUTS:
;3497     3566  2 !
;3498     3567  2 !       none
;3499     3568  2 !
;3500     3569  2 ! IMPLICIT OUTPUTS:
;3501     3570  2 !
;3502     3571  2 !       none
;3503     3572  2 !
;3504     3573  2 ! ROUTINE VALUE:
;3505     3574  2 !
;3506     3575  2 !       none
;3507     3576  2 !
;3508     3577  2 ! SIDE EFFECTS:
;3509     3578  2 !
;3510     3579  2 !       none
;3511     3580  2 !--
;3512     3581  2
;3513     3582  2 !
;3514     3583  2 ! Display strings.
;3515     3584  2 !
;3516     3585  2
;3517     3586  2 local
;3518     3587  2     status      : long ;
;3519     3588  2
;3520     3589  2 bind
;3521     3590  2     username    = cstring ('Username: !32AF Owner:  !AC'),
;3522     3591  2     account     = cstring ('Account:  !32AF UIC:    !%U (!%I)'),
;3523     3592  2     cli_table   = cstring ('CLI:      !32AC Tables: !AC'),
;3524     3593  2     default     = cstring ('Default:  !AC!AC'),
;3525     3594  2     lgicmd      = cstring ('LGICMD:   !AC'),
;3526     3595  2     flags       = cstring ('Login Flags: !AD'),
;3527     3596  2     flag_pad    = cstring (%char (cr), %char (lf), '                '),
;3528     3597  2     primdays    = cstring ('Primary days:  !7(AC)'),
;3529     3598  2     secdays     = cstring ('Secondary days:!7(AC)'),
;3530     3599  2     norestrict  = cstring ('No access restrictions'),
;3531   P 3600  2     accesshdr1  = cstring (
;3532     3601  2 'Primary   000000000011111111112222  Secondary 000000000011111111112222'),
;3533   P 3602  2     accesshdr2  = cstring (
;3534     3603  2 'Day Hours 012345678901234567890123  Day Hours 012345678901234567890123'),
```

```
;  3535          3604  2      netaccess   = cstring ('Network:  !AD            !AD'),
;  3536          3605  2      bataccess   = cstring ('Batch:    !AD            !AD'),
;  3537          3606  2      locaccess   = cstring ('Local:    !AD            !AD'),
;  3538          3607  2      diaaccess   = cstring ('Dialup:   !AD            !AD'),
;  3539          3608  2      remaccess   = cstring ('Remote:   !AD            !AD'),
;  3540          3609  2      expiration  = cstring ('Expiration: !AD    Pwdminimum: !2UL   Login Fails: !5UL'),
;  3541          3610  2      pwddata     = cstring ('Pwdlifetime:       !AD    Pwdchange:  !AD !AD'),
;  3542          3611  2      lastlogin   = cstring ('Last Login: !AD (interactive), !AD (non-interactive)'),
;  3543          3612  2      quota1      = cstring ('Maxjobs:     !5UL  Fillm:     !5UL  Bytlm:    !9UL'),
;  3544          3613  2      quota2      = cstring ('Maxacctjobs: !5UL  Shrfillm:  !5UL  Pbytlm:   !9UL'),
;  3545          3614  2      quota3      = cstring ('Maxdetach:   !5UL  BIOlm:     !5UL  JTquota:  !9UL'),
;  3546          3615  2      quota4      = cstring ('Prclm:       !5UL  DIOlm:     !5UL  WSdef:    !9UL'),
;  3547          3616  2      quota5      = cstring ('Prio:        !5UL  ASTlm:     !5UL  WSquo:    !9UL'),
;  3548          3617  2      quota6      = cstring ('Queprio:     !5UL  TQElm:     !5UL  WSextent: !9UL'),
;  3549          3618  2      quota7      = cstring ('CPU: !13AD  Enqlm:     !5UL  Pgflquo:  !9UL'),
;  3550          3619  2      privs       = cstring ('Authorized Privileges: '),
;  3551          3620  2      defprivs    = cstring ('Default Privileges: '),
;  3552          3621  2
;  3553          3622  2      nullstr     = cstring (''),
;  3554          3623  2
;  3555          3624  2      mon         = cstring (' Mon'),
;  3556          3625  2      tue         = cstring (' Tue'),
;  3557          3626  2      wed         = cstring (' Wed'),
;  3558          3627  2      thu         = cstring (' Thu'),
;  3559          3628  2      fri         = cstring (' Fri'),
;  3560          3629  2      sat         = cstring (' Sat'),
;  3561          3630  2      sun         = cstring (' Sun'),
;  3562          3631  2      noday       = cstring ('    '),
;  3563          3632  2
;  3564          3633  2      cputime     = recbuf[uaf$l_cputim]; ! CPU limit in hundredths of a second
;  3565          3634  2
;  3566          3635  2  own
;  3567          3636  2      flags_vector : vector [32] preset (
;  3568          3637  2      [$bitposition (uaf$v_audit)]        = cstring (' Audit'),
;  3569          3638  2      [$bitposition (uaf$v_captive)]      = cstring (' Captive'),
;  3570          3639  2      [$bitposition (uaf$v_defcli)]       = cstring (' Defcli'),
;  3571          3640  2      [$bitposition (uaf$v_disctly)]      = cstring (' Disctly'),
;  3572          3641  2      [$bitposition (uaf$v_nomail)]       = cstring (' Dismail'),
;  3573          3642  2      [$bitposition (uaf$v_dismail)]      = cstring (' Disnewmail'),
;  3574          3643  2      [$bitposition (uaf$v_disreconnect)] = cstring (' Disreconnect'),
;  3575          3644  2      [$bitposition (uaf$v_disreport)]    = cstring (' Disreport'),
;  3576          3645  2      [$bitposition (uaf$v_disacnt)]      = cstring (' Disuser'),
;  3577          3646  2      [$bitposition (uaf$v_diswelcom)]    = cstring (' Diswelcome'),
;  3578          3647  2      [$bitposition (uaf$v_genpwd)]       = cstring (' Genpwd'),
;  3579          3648  2      [$bitposition (uaf$v_lockpwd)]      = cstring (' Lockpwd'),
;  3580          3649  2      [$bitposition (uaf$v_pwd_expired)]  = cstring (' Pwd_expired'),
;  3581          3650  3      [$bitposition (uaf$v_pwd2_expired)] = cstring (' Pwd2_expired')
;  3582          3651  2      );
;  3583          3652  2
;  3584          3653  2  local
;  3585          3654  2          count,                              ! count for string being built
;  3586          3655  2          lcount,                             ! count of chars on current line
;  3587          3656  2          string          : vector [160, byte], ! buffer to build display string
;  3588          3657  2          flag_string     : ref vector [,byte], ! pointer to flag string
;  3589          3658  2          delta_time      : vector [long, 2],   ! Scratch area for system delta time
;  3590          3659  2          PTR,                                ! Pointer into UAF$Q_PWD_DATE quadword
;  3591          3660  2          time1           : vector [17, byte],  ! buffer for time string
```

```
;  3592          3661  2          time2           : vecto. [17, byte],    ! buffer for time string
;  3593          3662  2          time3           : vector [17, byte];    ! buffer for time string
;  3594          3663  2
;  3595          3664  2 builtin
;  3596          3665  2          emul;
;  3597          3666  2
;  3598          3667  2
;  3599          3668  2
;  3600          3669  2 if .str_wild and not fmg$match_name (namelen (uaf$s_username,recbuf[uaf$t_username]),
;  3601          3670  2                                     recbuf[uaf$t_username],
;  3602          3671  2                                     .match_token[en,  match_token)
;  3603          3672  2 then return true;
;  3604          3673  2
;  3605          3674  2 found_match = true;
;  3606          3675  2
;  3607          3676  2 output_null;
;  3608          3677  2
;  3609        P 3678  2 faomac (username,
;  3610        P 3679  2     uaf$s_username, recbuf[uaf$t_username],
;  3611          3680  2     recbuf[uaf$t_owner]);
;  3612          3681  2
;  3613        P 3682  2 faomac (account,
;  3614        P 3683  2     uaf$s_account, recbuf[uaf$t_account],
;  3615        P 3684  2     .recbuf[uaf$l_uic],
;  3616          3685  2     .recbuf[uaf$l_uic]);
;  3617          3686  2
;  3618        P 3687  2 faomac (cli_table,
;  3619        P 3688  2     recbuf[uaf$t_defcli],
;  3620          3689  2     recbuf[uaf$t_clitables]);
;  3621          3690  2
;  3622        P 3691  2 faomac (default,
;  3623        P 3692  2     recbuf[uaf$t_defdev],
;  3624          3693  2     recbuf[uaf$t_defdir]);
;  3625          3694  2
;  3626        P 3695  2 faomac (lgicmd,
;  3627          3696  2     recbuf[uaf$t_lgicmd]);
;  3628          3697  2
;  3629          3698  2 count = 0;
;  3630          3699  2 lcount = .flags<0,8>;
;  3631          3700  2 incr j from 0 to 31
;  3632          3701  2 do
;  3633          3702  3     begin
;  3634          3703  3     if .bitvector [recbuf[uaf$l_flags], .j]
;  3635          3704  3     and (flag_string = .flags_vector [.j]) neq 0
;  3636          3705  3     then
;  3637          3706  4         begin
;  3638          3707  4         if .lcount + .flag_string[0] gtru 80
;  3639          3708  4         then
;  3640          3709  5             begin
;  3641          3710  5             ch$move (.flag_pad<0,8>, flag_pad+1, string[.count]);
;  3642          3711  5             count = .count + .flag_pad<0,8>;
;  3643          3712  5             lcount = .flag_pad<0,8> - 2;
;  3644          3713  4             end;
;  3645          3714  4         ch$move (.flag_string[0], flag_string[1], string[.count]);
;  3646          3715  4         count = .count + .flag_string[0];
;  3647          3716  4         lcount = .lcount + .flag_string[0];
;  3648          3717  3         end;
```

```
;  3649          3718  2      end;
;  3650          3719  2
;  3651        P 3720  2  faomac (flags,
;  3652          3721  2      .count, string);
;  3653          3722  2
;  3654        P 3723  2  faomac (primdays,
;  3655        P 3724  2      if .recbuf[uaf$v_monday]      then noday   else mon,
;  3656        P 3725  2      if .recbuf[uaf$v_tuesday]     then noday   else tue,
;  3657        P 3726  2      if .recbuf[uaf$v_wednesday]   then noday   else wed,
;  3658        P 3727  2      if .recbuf[uaf$v_thursday]    then noday   else thu,
;  3659        P 3728  2      if .recbuf[uaf$v_friday]      then noday   else fri,
;  3660        P 3729  2      if .recbuf[uaf$v_saturday]    then noday   else sat,
;  3661          3730  2      if .recbuf[uaf$v_sunday]      then noday   else sun);
;  3662          3731  2
;  3663        P 3732  2  faomac (secdays,
;  3664        P 3733  2      if .recbuf[uaf$v_monday]      then mon     else noday,
;  3665        P 3734  2      if .recbuf[uaf$v_tuesday]     then tue     else noday,
;  3666        P 3735  2      if .recbuf[uaf$v_wednesday]   then wed     else noday,
;  3667        P 3736  2      if .recbuf[uaf$v_thursday]    then thu     else noday,
;  3668        P 3737  2      if .recbuf[uaf$v_friday]      then fri     else noday,
;  3669        P 3738  2      if .recbuf[uaf$v_saturday]    then sat     else noday,
;  3670          3739  2      if .recbuf[uaf$v_sunday]      then sun     else noday);
;  3671          3740  2
;  3672          3741  2  if ch$fail (ch$find_not_ch (10*uaf$s_network_access_p, recbuf[uaf$b_network_access_p], 0))
;  3673          3742  2  then
;  3674          3743  3      begin
;  3675          3744  3      faomac (norestrict);
;  3676          3745  3      end
;  3677          3746  2  else
;  3678          3747  3      begin
;  3679          3748  3      faomac (accesshdr1);
;  3680          3749  3      faomac (accesshdr2);
;  3681          3750  3
;  3682          3751  3      display_hours (netaccess, recbuf[uaf$b_network_access_p]);
;  3683          3752  3      display_hours (bataccess, recbuf[uaf$b_batch_access_p]);
;  3684          3753  3      display_hours (locaccess, recbuf[uaf$b_local_access_p]);
;  3685          3754  3      display_hours (diaaccess, recbuf[uaf$b_dialup_access_p]);
;  3686          3755  3      display_hours (remaccess, recbuf[uaf$b_remote_access_p]);
;  3687          3756  2      end;
;  3688          3757  2
;  3689          3758  2  convert_time (recbuf[uaf$q_expiration], 17, time1);
;  3690        P 3759  2  faomac (expiration,
;  3691        P 3760  2      17, time1,
;  3692        P 3761  2      .recbuf[uaf$b_pwd_length],
;  3693          3762  2      .recbuf[uaf$w_logfails]);
;  3694          3763  2
;  3695          3764  2  convert_time (recbuf[uaf$q_pwd_lifetime], 10, time1);
;  3696          3765  2  PTR = RECBUF[UAF$Q_PWD_DATE];   ! Because quadwords have "no" width
;  3697          3766  2  if (..PTR eql -1) and (.(.PTR+%upval) eql -1) then
;  3698          3767  2     ch$move(17, uplit(%ascii'    (pre-expired)'), TIME2)
;  3699          3768  2  else
;  3700          3769  2     CONVERT_TIME(.PTR, 17, TIME2);
;  3701          3770  2  convert_time (recbuf[uaf$q_pwd2_date], 17, time3);
;  3702        P 3771  2  faomac (pwddata,
;  3703        P 3772  2      10, time1,
;  3704        P 3773  2      17, time2,
;  3705        P 3774  2      (if (. (recbuf[uaf$q_pwd2_date]+0) or . (recbuf[uaf$q_pwd2_date]+4)) eql 0
```

```
;  3706        P 3775  2       then 0
;  3707          3776  2       else 17), time3);
;  3708          3777  2
;  3709          3778  2 convert_time (recbuf[uaf$q_lastlogin_i], 17, time1);
;  3710          3779  2 convert_time (recbuf[uaf$q_lastlogin_n], 17, time2);
;  3711        P 3780  2 faomac (lastlogin,
;  3712        P 3781  2     17, time1,
;  3713          3782  2     17, time2);
;  3714          3783  2
;  3715          3784  2 emul (%ref (-200000), %ref (.cputime<1,31>),
;  3716          3785  2       %ref (if .cputime<0,1> then -100000 else 0), delta_time);
;  3717          3786  2 convert_time (delta_time, 13, time1);
;  3718          3787  2
;  3719        P 3788  2 faomac (quota1,
;  3720        P 3789  2     .recbuf[uaf$w_maxjobs],
;  3721        P 3790  2     .recbuf[uaf$w_fillm],
;  3722          3791  2     .recbuf[uaf$l_bytlm]);
;  3723          3792  2
;  3724        P 3793  2 faomac (quota2,
;  3725        P 3794  2     .recbuf[uaf$w_maxacctjobs],
;  3726        P 3795  2     .recbuf[uaf$w_shrfillm],
;  3727          3796  2     .recbuf[uaf$l_pbytlm]);
;  3728          3797  2
;  3729        P 3798  2 faomac (quota3,
;  3730        P 3799  2     .recbuf[uaf$w_maxdetach],
;  3731        P 3800  2     .recbuf[uaf$w_biolm],
;  3732          3801  2     .recbuf[uaf$l_jtquota]);
;  3733          3802  2
;  3734        P 3803  2 faomac (quota4,
;  3735        P 3804  2     .recbuf[uaf$w_prccnt],
;  3736        P 3805  2     .recbuf[uaf$w_diolm],
;  3737          3806  2     .recbuf[uaf$l_dfwscnt]);
;  3738          3807  2
;  3739        P 3808  2 faomac (quota5,
;  3740        P 3809  2     .recbuf[uaf$b_pri],
;  3741        P 3810  2     .recbuf[uaf$w_astlm],
;  3742          3811  2     .recbuf[uaf$l_wsquota]);
;  3743          3812  2
;  3744        P 3813  2 faomac (quota6,
;  3745        P 3814  2     .recbuf[uaf$b_quepri],
;  3746        P 3815  2     .recbuf[uaf$w_tqcnt],
;  3747          3816  2     .recbuf[uaf$l_wsextent]);
;  3748          3817  2
;  3749        P 3818  2 faomac (quota7,
;  3750        P 3819  2     13, time1,
;  3751        P 3820  2     .recbuf[uaf$w_englm],
;  3752          3821  2     .recbuf[uaf$l_pgflquota]);
;  3753          3822  2
;  3754          3823  2 faomac (privs);
;  3755          3824  2 print_priv (recbuf[uaf$q_priv]);
;  3756          3825  2
;  3757          3826  2 faomac (defprivs);
;  3758          3827  2 print_priv (recbuf[uaf$q_def_priv]);
;  3759          3828  2
;  3760          3829  2 !
;  3761          3830  2 ! Build a holder from the UAF record and display the rights
;  3762          3831  2 ! granted to it.
```

```
;  3763          3832  2 !
;  3764          3833  2 if .rdb_exists
;  3765          3834  2 then
;  3766          3835  3     begin
;  3767          3836  3     uaf$build_holder ();
;  3768          3837  3     rdb_header_flag = true ;
;  3769          3838  3     status = uaf$write_rights ( holder ) ;
;  3770          3839  2     end ;
;  3771          3840  2
;  3772          3841  2 return .status ;
;  3773          3842  1 end;


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                        1B 0032D P.ACY:  .BYTE   27
46 41 32 33 21 20 3A 65 6D 61 6E 72 65 73               55 0032E         .ASCII  \Username: !32AF Owner:  !AC\
                  43 41 21 20 20 3A 72 65 6E 77 4F      20 0033D
                                                        21 00349 P.ACZ:  .BYTE   33
46 41 32 33 21 20 20 3A 74 6E 75 6F 63 63               41 0034A         .ASCII  \Account:  !32AF UIC:    !%U (!%I)\
21 28 20 55 25 21 20 20 20 20 3A 43 49 55               20 00359
                              29 49                     25 00368
                                                        1B 0036B P.ADA:  .BYTE   27
43 41 32 33 21 20 20 20 20 20 20 3A 49 4C               43 0036C         .ASCII  \CLI:       !32AC Tables: !AC\
                  43 41 21 20 3A 73 65 6C 62 61 54      20 0037B
                                                        10 00387 P.ADB:  .BYTE   16
41 21 43 41 21 20 20 3A 74 6C 75 61 66 65               44 00388         .ASCII  \Default:  !AC!AC\
                                                        43 00397
                                                        0D 00398 P.ADC:  .BYTE   13
         43 41 21 20 20 20 3A 44 4D 43 49 47            4C 00399         .ASCII  \LGICMD:   !AC\
                                                        10 003A6 P.ADD:  .BYTE   16
41 21 20 3A 73 67 61 6C 46 20 6E 69 67 6F               4C 003A7         .ASCII  \Login Flags: !AD\
                                                        44 003B6
                                                        0F 003B7 P.ADE:  .BYTE   15
20 20 20 20 20 20 20 20 20 20 20 20 20 0A               0D 003B8         .ASCII  <13><10>\             \
                                                        15 003C7 P.ADF:  .BYTE   21
20 20 3A 73 79 61 64 20 79 72 61 6D 69 72               50 003C8         .ASCII  \Primary days:  !7(AC)\
                        29 43 41 28 37                  21 003D7
                                                        15 003DD P.ADG:  .BYTE   21
3A 73 79 61 64 20 79 72 61 64 6E 6F 63 65               53 003DE         .ASCII  \Secondary days:!7(AC)\
                        29 43 41 28 37                  21 003ED
                                                        16 003F3 P.ADH:  .BYTE   22
72 74 73 65 72 20 73 73 65 63 63 61 20 6F               4E 003F4         .ASCII  \No access restrictions\
                        73 6E 6F 69 74 63               69 00403
                                                        46 0040A P.ADI:  .BYTE   70
30 30 30 30 30 20 20 20 79 72 61 6D 69 72               50 0040B         .ASCII  \Primary   000000000011111111112222  Seco\
31 31 31 31 31 31 31 31 31 31 30 30 30 30               30 0041A
                  6F 63 65 53 20 20 32 32 32 32         32 00429
30 30 30 30 30 30 30 30 20 79 72 61 64 6E               6E 00433         .ASCII  \ndary 000000000011111111112222\
32 32 32 32 31 31 31 31 31 31 31 31 31 31               30 00442
                                                        46 00451 P.ADJ:  .BYTE   70
34 33 32 31 30 20 73 72 75 6F 48 20 79 61               44 00452         .ASCII  \Day Hours 012345678901234567890123  Day \
39 38 37 36 35 34 33 32 31 30 39 38 37 36               35 00461
                  20 79 61 44 20 20 33 32 31 30         30 00470
38 37 36 35 34 33 32 31 30 20 73 72 75 6F               48 0047A         .ASCII  \Hours 012345678901234567890123\
33 32 31 30 39 38 37 36 35 34 33 32 31 30               39 00489
```

```
                                                
```

```
20 20 20 20 20 3A 6D 6C 45 51 54 20 20 4C 55 006D3
               78 65 53 57 20 20 4C 55 35 21 006E2
               4C 55 39 21 20 3A 74 6E 65 74 006EC          .ASCII  \tent: !9UL\
                                          2B 006F6 P.ADY:   .BYTE   43
71 6E 45 20 20 44 41 33 31 21 20 3A 55 50 43 006F7          .ASCII  \CPU: !13AD  Enqlm:     !5UL  Pgflquo:  !\
50 20 20 4C 55 35 21 20 20 20 20 20 3A 6D 6C 00706
               21 20 20 3A 6F 75 71 6C 66 67 00715
                                  4C 55 39 0071F          .ASCII  \9UL\
                                          17 00722 P.ADZ:   .BYTE   23
76 69 72 50 20 64 65 7A 69 72 6F 68 74 75 41 00723          .ASCII  \Authorized Privileges: \
                        20 3A 73 65 67 65 6C 69 00732
                                          14 0073A P.AEA:   .BYTE   20
65 6C 69 76 69 72 50 20 74 6C 75 61 66 65 44 0073B          .ASCII  \Default Privileges: \
                              20 3A 73 65 67 0074A
                                          00 0074F P.AEB:   .BYTE   0
                                             00750          .BLKB   0
                                          04 00750 P.AEC:   .BYTE   4
                                 6E 6F 4D 20 00751          .ASCII  \ Mon\
                                          04 00755 P.AED:   .BYTE   4
                                 65 75 54 20 00756          .ASCII  \ Tue\
                                          04 0075A P.AEE:   .BYTE   4
                                 64 65 57 20 0075B          .ASCII  \ Wed\
                                          04 0075F P.AEF:   .BYTE   4
                                 75 68 54 20 00760          .ASCII  \ Thu\
                                          04 00764 P.AEG:   .BYTE   4
                                 69 72 46 20 00765          .ASCII  \ Fri\
                                          04 00769 P.AEH:   .BYTE   4
                                 74 61 53 20 0076A          .ASCII  \ Sat\
                                          04 0076E P.AEI:   .BYTE   4
                                 6E 75 53 20 0076F          .ASCII  \ Sun\
                                          04 00773 P.AEJ:   .BYTE   4
                                 20 20 20 20 00774          .ASCII  \    \
                                          06 00778 P.AEK:   .BYTE   6
                              74 69 64 75 41 20 00779       .ASCII  \ Audit\
                                          08 0077F P.AEL:   .BYTE   8
                           65 76 69 74 70 61 43 20 00780    .ASCII  \ Captive\
                                          07 00788 P.AEM:   .BYTE   7
                              69 6C 63 66 65 44 20 00789    .ASCII  \ Defcli\
                                          08 00790 P.AEN:   .BYTE   8
                           79 6C 74 63 73 69 44 20 00791    .ASCII  \ Disctly\
                                          08 00799 P.AEO:   .BYTE   8
                           6C 69 61 6D 73 69 44 20 0079A    .ASCII  \ Dismail\
                                          0B 007A2 P.AEP:   .BYTE   11
                     6C 69 61 6D 77 65 6E 73 69 44 20 007A3 .ASCII  \ Disnewmail\
                                          0D 007AE P.AEQ:   .BYTE   13
74 63 65 6E 6E 6F 63 65 72 73 69 44 20 007AF                .ASCII  \ Disreconnect\
                                          0A 007BC P.AER:   .BYTE   10
                        74 72 6F 70 65 72 73 69 44 20 0078D .ASCII  \ Disreport\
                                          08 007C7 P.AES:   .BYTE   8
                           72 65 73 75 73 69 44 20 007C8    .ASCII  \ Disuser\
                                          0B 007D0 P.AET:   .BYTE   11
                     65 6D 6F 63 6C 65 77 73 69 44 20 007D1 .ASCII  \ Diswelcome\
                                          07 007DC P.AEU:   .BYTE   7
                              64 77 70 6E 65 47 20 007DD    .ASCII  \ Genpwd\
                                          08 007E4 P.AEV:   .BYTE   8
                           64 77 70 6B 63 6F 4C 20 007E5    .ASCII  \ Lockpwd\
                                          0C 007ED P.AEW:   .BYTE   12
```

```
                64  65  72  69  70  78  65  5F  64  77  50  20  007EE           .ASCII  \ Pwd_expired\                                  ;
                                                        0D  007FA P.AEX:    .BYTE   13                                               ;
            64  65  72  69  70  78  65  5F  32  64  77  50  20  007FB           .ASCII  \ Pwd2_expired\                                 ;
65  72  69  70  78  65  2D  65  72  70  28  20  20  20  20  00808 P.AEY:    .ASCII  \    (pre-expired)\<0><0><0>                    ;
                                            00  00  00  29  64  00817                                                               ;

                                                                                .PSECT  $OWN$,NOEXE,2

00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 0108C FLAGS_VECTOR:
                                                                                .ADDRESS P.AEN, P.AEM, P.AEV, P.AEL, P.AES, -           ;
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 010A4                       P.AET, P.AEP, P.AEO, P.AEU, P.AEW, P.AEX, - ;
                                        00000000' 00000000' 010BC                       P.AEK, P.AER, P.AEQ                         ;
                                                            010C4               .BLKB   72

                                                                    USERNAME=           P.ACY
                                                                    ACCOUNT=            P.ACZ
                                                                    CLI_TABLE=          P.ADA
                                                                    DEFAULT=            P.ADB
                                                                    LGICMD=             P.ADC
                                                                    FLAGS=              P.ADD
                                                                    FLAG_PAD=           P.ADE
                                                                    PRIMDAYS=           P.ADF
                                                                    SECDAYS=            P.ADG
                                                                    NORESTRICT=         P.ADH
                                                                    ACCESSHDR1=         P.ADI
                                                                    ACCESSHDR2=         P.ADJ
                                                                    NETACCESS=          P.ADK
                                                                    BATACCESS=          P.ADL
                                                                    LOCACCESS=          P.ADM
                                                                    DIAACCESS=          P.ADN
                                                                    REMACCESS=          P.ADO
                                                                    EXPIRATION=         P.ADP
                                                                    PWDDATA=            P.ADQ
                                                                    LASTLOGIN=          P.ADR
                                                                    QUOTA1=             P.ADS
                                                                    QUOTA2=             P.ADT
                                                                    QUOTA3=             P.ADU
                                                                    QUOTA4=             P.ADV
                                                                    QUOTA5=             P.ADW
                                                                    QUOTA6=             P.ADX
                                                                    QUOTA7=             P.ADY
                                                                    PRIVS=              P.ADZ
                                                                    DEFPRIVS=           P.AEA
                                                                    NULLSTR=            P.AEB
                                                                    MON=                P.AEC
                                                                    TUE=                P.AED
                                                                    WED=                P.AEE
                                                                    THU=                P.AEF
                                                                    FRI=                P.AEG
                                                                    SAT=                P.AEH
                                                                    SUN=                P.AEI
                                                                    NODAY=              P.AEJ
                                                                    CPUTIME=            RECBUF+556

                                                                                .PSECT  $CODE$,NOWRT,2
```

```
                                    OFFC 00000 DISPLAY_FULL:
                                                 .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 3548
                          5E     FF1C     CE  9E 00002         MOVAB   -228(SP), SP
                          31 00000000'    00  E9 00007         BLBC    STR_WILD, 1$                     ; 3669
                          55 00000000'    00  9E 0000E         MOVAB   MATCH_TOKEN, R5                  ; 3670
                          53 00000000'    00  9E 00015         MOVAB   RECBUF+4, R3
00000000'  00             20              20  3A 0001C         LOCC    #32, #32, RECBUF+4               ; 3669
                          52        E0    A0  9E 00024         MOVAB   -32(R0), R2
                          52              52  CE 00028         MNEGL   R2, R2
                          54 00000000'    00  D0 0002B         MOVL    MATCH_TOKENLEN, R4               ; 3670
                             00000000G    00  16 00032         JSB     FMG$MATCH_NAME
                          04              50  E8 00038         BLBS    R0, 1$
                          50              01  D0 0003B         MOVL    #1, R0                           ; 3672
                                          04  0003E         RET
                00000000'  00             01  D0 0003F 1$:    MOVL    #1, FOUND_MATCH                  ; 3674
                          50 00000000'    00  D0 00046         MOVL    RABPTR, R0
                                    22    A0  B4 0004D         CLRW    34(R0)
                                          50  DD 00050         PUSHL   R0
                00000000G  00             01  FB 00052         CALLS   #1, SYS$PUT
                00000000'  00 00000000'   00  9B 00059         MOVZBW  USERNAME, FAODSC                 ; 3680
                00000000'  00 00000000'   00  9E 00064         MOVAB   USERNAME+1, FAODSC+4
                           00000000'      00  9F 0006F         PUSHAB  RECBUF+84
                           00000000'      00  9F 00075         PUSHAB  RECBUF+4
                                          20  DD 0007B         PUSHL   #32
                           00000000'      00  9F 0007D         PUSHAB  DISDSC
             7E 00000000'  00             22  C1 00083         ADDL3   #34, RABPTR, -(SP)
                           00000000'      00  9F 0008B         PUSHAB  FAODSC
                00000000G  00             06  FB 00091         CALLS   #6, SYS$FAO
                           00000000'      00  DD 00098         PUSHL   RABPTR
                00000000G  00             01  FB 0009E         CALLS   #1, SYS$PUT
                00000000'  00 00000000'   00  9B 000A5         MOVZBW  ACCOUNT, FAODSC                  ; 3685
                00000000'  00 00000000'   00  9E 000B0         MOVAB   ACCOUNT+1, FAODSC+4
                           50 00000000'   00  D0 000BB         MOVL    RECBUF+36, R0
                                          50  DD 000C2         PUSHL   R0
                                          50  DD 000C4         PUSHL   R0
                           00000000'      00  9F 000C6         PUSHAB  RECBUF+52
                                          20  DD 000CC         PUSHL   #32
                           00000000'      00  9F 000CE         PUSHAB  DISDSC
             7E 00000000'  00             22  C1 000D4         ADDL3   #34, RABPTR, -(SP)
                           00000000'      00  9F 000DC         PUSHAB  FAODSC
                00000000G  00             07  FB 000E2         CALLS   #7, SYS$FAO
                           00000000'      00  DD 000E9         PUSHL   RABPTR
                00000000G  00             01  FB 000EF         CALLS   #1, SYS$PUT
                00000000'  00 00000000'   00  9B 000F6         MOVZBW  CLI_TABLE, FAODSC                ; 3689
                00000000'  00 00000000'   00  9E 00101         MOVAB   CLI_TABLE+1, FAODSC+4
                           00000000'      00  9F 0010C         PUSHAB  RECBUF+308
                           00000000'      00  9F 00112         PUSHAB  RECBUF+276
                           00000000'      00  9F 00118         PUSHAB  DISDSC
             7E 00000000'  00             22  C1 0011E         ADDL3   #34, RABPTR, -(SP)
                           00000000'      00  9F 00126         PUSHAB  FAODSC
                00000000G  00             05  FB 0012C         CALLS   #5, SYS$FAO
                           00000000'      00  DD 00133         PUSHL   RABPTR
                00000000G  00             01  FB 00139         CALLS   #1, SYS$PUT
                00000000'  00 00000000'   00  9B 00140         MOVZBW  DEFAULT, FAODSC                  ; 3693
                00000000'  00 00000000'   00  9E 0014B         MOVAB   DEFAULT+1, FAODSC+4
                           00000000'      00  9F 00156         PUSHAB  RECBUF+148
                           00000000'      00  9F 0015C         PUSHAB  RECBUF+116
```

```
                                00000000'  00  9F 00162          PUSHAB  DISDSC                                                   ;
                      7E 00000000'  00         22 C1 00168       ADDL3   #34, RABPTR, -(SP)                                      ;
                                00000000'  00  9F 00170          PUSHAB  FAODSC                                                  ;
                         00000000G  00         05 FB 00176       CALLS   #5, SYS$FAO                                             ;
                                00000000'  00  DD 0017D          PUSHL   RABPTR                                                  ;
                         00000000G  00         01 FB 00183       CALLS   #1, SYS$PUT                                             ;
                         00000000'  00 00000000'  00  9B 0018A    MOVZBW  LGICMD, FAODSC                                          ; 3696
                         00000000'  00 00000000'  00  9E 00195    MOVAB   LGICMD+1, FAODSC+4                                      ;
                                00000000'  00  9F 001A0          PUSHAB  RECBUF+212                                              ;
                                00000000'  00  9F 001A6          PUSHAB  DISDSC                                                  ;
                      7E 00000000'  00         22 C1 001AC       ADDL3   #34, RABPTR, -(SP)                                      ;
                                00000000'  00  9F 001B4          PUSHAB  FAODSC                                                  ;
                         00000000G  00         04 FB 001BA       CALLS   #4, SYS$FAO                                             ;
                                00000000'  00  DD 001C1          PUSHL   RABPTR                                                  ;
                         00000000G  00         01 FB 001C7       CALLS   #1, SYS$PUT                                             ;
                                               57 D4 001CE       CLRL    COUNT                                                   ; 3698
                                5B 00000000'   00 9A 001D0       MOVZBL  FLAGS, R11                                              ; 3699
                                56             5B D0 001D7       MOVL    R11, LCOUNT                                             ;
                                               59 D4 001DA       CLRL    J                                                       ; 3700
                      47 00000000'  00         59 E1 001DC 2$:   BBC     J, RECBUF+468, 4$                                       ; 3703
                                5A 00000000'0049 D0 001E4        MOVL    FLAGS_VECTOR[J], FLAG_STRING                            ; 3704
                                               3D 13 001EC       BEQL    4$                                                      ;
                                58             6A 9A 001EE       MOVZBL  (FLAG_STRING), R8                                       ; 3707
                                58             56 C0 001F1       ADDL2   LCOUNT, R8                                              ;
                         00000050   8F         58 D1 001F4       CMPL    R8, #80                                                 ;
                                               18 1B 001FB       BLEQU   3$                                                      ;
                                58 00000000'   00 9A 001FD       MOVZBL  FLAG_PAD, R8                                            ; 3710
                   44 AE47 00000000'  00       58 28 00204       MOVC3   R8, FLAG_PAD+1, STRING[COUNT]                           ;
                                57             58 C0 0020E       ADDL2   R8, COUNT                                               ; 3711
                                56       FE    A8 9E 00211       MOVAB   -2(R8), LCOUNT                                          ; 3712
                                50             6A 9A 00215 3$:   MOVZBL  (FLAG_STRING), R0                                       ; 3714
             44 AE47       01    AA            50 28 00218       MOVC3   R0, @(FLAG_STRING), STRING[COUNT]                       ;
                                50             6A 9A 0021F       MOVZBL  (FLAG_STRING), R0                                       ; 3715
                                57             50 C0 00222       ADDL2   R0, COUNT                                               ;
                                50             6A 9A 00225       MOVZBL  (FLAG_STRING), R0                                       ; 3716
                                56             50 C0 00228       ADDL2   R0, LCOUNT                                              ;
                      AD                       1F F3 0022B 4$:   AOBLEQ  #31, J, 2$                                              ; 3700
                         00000000'  00         5B B0 0022F       MOVW    R11, FAODSC                                             ; 3721
                         00000000'  00 00000000'  00  9E 00236    MOVAB   FLAGS+1, FAODSC+4                                       ;
                                      44       AE 9F 00241       PUSHAB  STRING                                                  ;
                                               57 DD 00244       PUSHL   COUNT                                                   ;
                                00000000'      00 9F 00246       PUSHAB  DISDSC                                                  ;
                      7E 00000000'  00         22 C1 0024C       ADDL3   #34, RABPTR, -(SP)                                      ;
                                00000000'      00 9F 00254       PUSHAB  FAODSC                                                  ;
                         00000000G  00         05 FB 0025A       CALLS   #5, SYS$FAO                                             ;
                                00000000'      00 DD 00261       PUSHL   RABPTR                                                  ;
                         00000000G  00         01 FB 00267       CALLS   #1, SYS$PUT                                             ;
                         00000000'  00 00000000'  00  9B 0026E    MOVZBW  PRIMDAYS, FAODSC                                        ; 3730
                         00000000'  00 00000000'  00  9E 00279    MOVAB   PRIMDAYS+1, FAODSC+4                                    ;
                      09 00000000'  00         06 E1 00284       BBC     #6, RECBUF+514, 5$                                      ;
                                50 00000000'   00 9E 0028C       MOVAB   NODAY, R0                                               ;
                                               07 11 00293       BRB     6$                                                      ;
                                50 00000000'   00 9E 00295 5$:   MOVAB   SUN, R0                                                 ;
                                               50 DD 0029C 6$:   PUSHL   R0                                                      ;
                      09 00000000'  00         05 E1 0029E       BBC     #5, RECBUF+514, 7$                                      ;
                                50 00000000'   00 9E 002A6       MOVAB   NODAY, R0                                               ;
                                               07 11 002AD       BRB     8$                                                      ;
```

```
                                50 00000000'  00  9E 002AF 7$:      MOVAB   SAT, R0                                 ;
                                              50  DD 002B6 8$:      PUSHL   R0                                      ;
                          09 00000000'  00    04  E1 002B8         BBC     #4, RECBUF+514, 9$                      ;
                                50 00000000'  00  9E 002C0         MOVAB   NODAY, R0                               ;
                                              07  11 002C7         BRB     10$                                     ;
                                50 00000000'  00  9E 002C9 9$:      MOVAB   FRI, R0                                 ;
                                              50  DD 002D0 10$:     PUSHL   R0                                      ;
                          09 00000000'  00    03  E1 002D2         BBC     #3, RECBUF+514, 11$                     ;
                                50 00000000'  00  9E 002DA         MOVAB   NODAY, R0                               ;
                                              07  11 002E1         BRB     12$                                     ;
                                50 00000000'  00  9E 002E3 11$:     MOVAB   THU, R0                                 ;
                                              50  DD 002EA 12$:     PUSHL   R0                                      ;
                          09 00000000'  00    02  E1 002EC         BBC     #2, RECBUF+514, 13$                     ;
                                50 00000000'  00  9E 002F4         MOVAB   NODAY, R0                               ;
                                              07  11 002FB         BRB     14$                                     ;
                                50 00000000'  00  9E 002FD 13$:     MOVAB   WED, R0                                 ;
                                              50  DD 00304 14$:     PUSHL   R0                                      ;
                          09 00000000'  00    01  E1 00306         BBC     #1, RECBUF+514, 15$                     ;
                                50 00000000'  00  9E 0030E         MOVAB   NODAY, R0                               ;
                                              07  11 00315         BRB     16$                                     ;
                                50 00000000'  00  9E 00317 15$:     MOVAB   TUE, R0                                 ;
                                              50  DD 0031E 16$:     PUSHL   R0                                      ;
                                09 00000000'  00  E9 00320         BLBC    RECBUF+514, 17$                         ;
                                50 00000000'  00  9E 00327         MOVAB   NODAY, R0                               ;
                                              07  11 0032E         BRB     18$                                     ;
                                50 00000000'  00  9E 00330 17$:     MOVAB   MON, R0                                 ;
                                              50  DD 00337 18$:     PUSHL   R0                                      ;
                                   00000000'  00  9F 00339         PUSHAB  DISDSC                                  ;
                       7E 00000000'  00       22  C1 0033F         ADDL3   #34, RABPTR, -(SP)                      ;
                                   00000000'  00  9F 00347         PUSHAB  FAODSC                                  ;
                          00000000G  00       0A  FB 0034D         CALLS   #10, SYS$FAO                            ;
                                   00000000'  00  DD 00354         PUSHL   RABPTR                                  ;
                          00000000G  00       01  FB 0035A         CALLS   #1, SYS$PUT                             ;
                          00000000'  00 00000000'  00  9B 00361     MOVZBW  SECDAYS, FAODSC                         ; 3739
                          00000000'  00 00000000'  00  9E 0036C     MOVAB   SECDAYS+1, FAODSC+4                     ;
                    09 00000000'  00          06  E1 00377         BBC     #6, RECBUF+514, 19$                     ;
                                50 00000000'  00  9E 0037F         MOVAB   SUN, R0                                 ;
                                              07  11 00386         BRB     20$                                     ;
                                50 00000000'  00  9E 00388 19$:     MOVAB   NODAY, R0                               ;
                                              50  DD 0038F 20$:     PUSHL   R0                                      ;
                    09 00000000'  00          05  E1 00391         BBC     #5, RECBUF+514, 21$                     ;
                                50 00000000'  00  9E 00399         MOVAB   SAT, R0                                 ;
                                              07  11 003A0         BRB     22$                                     ;
                                50 00000000'  00  9E 003A2 21$:     MOVAB   NODAY, R0                               ;
                                              50  DD 003A9 22$:     PUSHL   R0                                      ;
                    09 00000000'  00          04  E1 003AB         BBC     #4, RECBUF+514, 23$                     ;
                                50 00000000'  00  9E 003B3         MOVAB   FRI, R0                                 ;
                                              07  11 003BA         BRB     24$                                     ;
                                50 00000000'  00  9E 003BC 23$:     MOVAB   NODAY, R0                               ;
                                              50  DD 003C3 24$:     PUSHL   R0                                      ;
                    09 00000000'  00          03  E1 003C5         BBC     #3, RECBUF+514, 25$                     ;
                                50 00000000'  00  9E 003CD         MOVAB   THU, R0                                 ;
                                              07  11 003D4         BRB     26$                                     ;
                                50 00000000'  00  9E 003D6 25$:     MOVAB   NODAY, R0                               ;
                                              50  DD 003DD 26$:     PUSHL   R0                                      ;
                    09 00000000'  00          02  E1 003DF         BBC     #2, RECBUF+514, 27$                     ;
                                50 00000000'  00  9E 003E7         MOVAB   WED, R0                                 ;
```

```
                                                07  11  003EE             BRB     28$
                                    50 00000000'00  9E  003F0 27$:        MOVAB   NODAY, R0
                                                50  DD  003F7 28$:        PUSHL   R0
                                09 00000000'00      01  E1  003F9         BBC     #1, RECBUF+514, 29$
                                    50 00000000'00  9E  00401             MOVAB   TUE, R0
                                                07  11  00408             BRB     30$
                                    50 00000000'00  9E  0040A 29$:        MOVAB   NODAY, R0
                                                50  DD  00411 30$:        PUSHL   R0
                                    09 00000000'00  E9  00413             BLBC    RECBUF+514, 31$
                                    50 00000000'00  9E  0041A             MOVAB   MON, R0
                                                07  11  00421             BRB     32$
                                    50 00000000'00  9E  00423 31$:        MOVAB   NODAY, R0
                                                50  DD  0042A 32$:        PUSHL   R0
                                       00000000'00  9F  0042C             PUSHAB  DISDSC
                                7E 00000000'    00      22  C1  00432     ADDL3   #34, RABPTR, -(SP)
                                       00000000'00  9F  0043A             PUSHAB  FAODSC
                                   00000000G    00      0A  FB  00440     CALLS   #10, SYS$FAO
                                       00000000'00  DD  00447             PUSHL   RABPTR
                                   00000000G    00      01  FB  0044D     CALLS   #1, SYS$PUT
                                52 00000000'    00      22  C1  00454     ADDL3   #34, RABPTR, R2                                 ;3744
                00000000'       00              1E      00  3B  0045C     SKPC    #0, #30, RECBUF+472                             ;3741
                                                02  12  00464             BNEQ    33$
                                                51  D4  00466             CLRL    R1
                                                51  D5  00468 33$:        TSTL    R1
                                                3B  12  0046A             BNEQ    34$
                                   00000000'    00 00000000'00  9B  0046C MOVZBW  NORESTRICT, FAODSC                               ;3744
                                   00000000'    00 00000000'00  9E  00477 MOVAB   NORESTRICT+1, FAODSC+4
                                       00000000'00  9F  00482             PUSHAB  DISDSC
                                                52  DD  00488             PUSHL   R2
                                       00000000'00  9F  0048A             PUSHAB  FAODSC
                                   00000000G    00      03  FB  00490     CALLS   #3, SYS$FAO
                                       00000000'00  DD  00497             PUSHL   RABPTR
                                   00000000G    00      01  FB  0049D     CALLS   #1, SYS$PUT
                                                00D5    31  004A4         BRW     35$                                             ;3741
                                   00000000'    00 00000000'00  9B  004A7 34$:    MOVZBW  ACCESSHDR1, FAODSC                      ;3748
                                   00000000'    00 00000000'00  9E  004B2 MOVAB   ACCESSHDR1+1, FAODSC+4
                                       00000000'00  9F  004BD             PUSHAB  DISDSC
                                                52  DD  004C3             PUSHL   R2
                                       00000000'00  9F  004C5             PUSHAB  FAODSC
                                   00000000G    00      03  FB  004CB     CALLS   #3, SYS$FAO
                                       00000000'00  DD  004D2             PUSHL   RABPTR
                                   00000000G    00      01  FB  004D8     CALLS   #1, SYS$PUT
                                   00000000'    00 00000000'00  9B  004DF MOVZBW  ACCESSHDR2, FAODSC                               ;3749
                                   00000000'    00 00000000'00  9E  004EA MOVAB   ACCESSHDR2+1, FAODSC+4
                                       00000000'00  9F  004F5             PUSHAB  DISDSC
                                7E 00000000'    00      22  C1  004FB     ADDL3   #34, RABPTR, -(SP)
                                       00000000'00  9F  00503             PUSHAB  FAODSC
                                   00000000G    00      03  FB  00509     CALLS   #3, SYS$FAO
                                       00000000'00  DD  00510             PUSHL   RABPTR
                                   00000000G    00      01  FB  00516     CALLS   #1, SYS$PUT
                                       00000000'00  9F  0051D             PUSHAB  RECBUF+472                                      ;3751
                                       00000000'00  9F  00523             PUSHAB  NETACCESS
                                   00000000V    00      02  FB  00529     CALLS   #2, DISPLAY_HOURS
                                       00000000'00  9F  00530             PUSHAB  RECBUF+478                                      ;3752
                                       00000000'00  9F  00536             PUSHAB  BATACCESS
                                   00000000V    00      02  FB  0053C     CALLS   #2, DISPLAY_HOURS
                                       00000000'00  9F  00543             PUSHAB  RECBUF+484                                      ;3753
```

```
                                                
```

```
                                00000000'  00  9F  00679          PUSHAB  FAODSC                                                    ;
                        00000000G  00          09  FB  0067F          CALLS   #9, SYS$FAO                                             ;
                                00000000'  00  DD  00686          PUSHL   RABPTR                                                  ;
                        00000000G  00          01  FB  0068C          CALLS   #1, SYS$PUT                                             ;
                                      28   AE  9F  00693          PUSHAB  TIME1                                                   ; 3778
                                           11  DD  00696          PUSHL   #17                                                     ;
                                00000000'  00  9F  00698          PUSHAB  RECBUF+396                                              ;
                        00000000V  00          03  FB  0069E          CALLS   #3, CONVERT_TIME                                        ;
                                      14   AE  9F  006A5          PUSHAB  TIME2                                                   ; 3779
                                           11  DD  006A8          PUSHL   #17                                                     ;
                                00000000'  00  9F  006AA          PUSHAB  RECBUF+404                                              ;
                        00000000V  00          03  FB  006B0          CALLS   #3, CONVERT_TIME                                        ;
                        00000000'  00  00000000'  00  9B  006B7          MOVZBW  LASTLOGIN, FAODSC                                       ; 3782
                        00000000'  00  00000000'  00  9E  006C2          MOVAB   LASTLOGIN+1, FAODSC+4                                   ;
                                      14   AE  9F  006CD          PUSHAB  TIME2                                                   ;
                                           11  DD  006D0          PUSHL   #17                                                     ;
                                      30   AE  9F  006D2          PUSHAB  TIME1                                                   ;
                                           11  DD  006D5          PUSHL   #17                                                     ;
                                00000000'  00  9F  006D7          PUSHAB  DISDSC                                                  ;
                     7E 00000000'  00          22  C1  006DD          ADDL3   #34, RABPTR, -(SP)                                      ;
                                00000000'  00  9F  006E5          PUSHAB  FAODSC                                                  ;
                        00000000G  00          07  FB  006EB          CALLS   #7, SYS$FAO                                             ;
                                00000000'  00  DD  006F2          PUSHL   RABPTR                                                  ;
                        00000000G  00          01  FB  006F8          CALLS   #1, SYS$PUT                                             ;
           51 00000000'  00                1F  01  EF  006FF          EXTZV   #1, #31, CPUTIME, R1                                    ; 3784
                                09 00000000'  00  E9  00708          BLBC    CPUTIME, 40$                                            ; 3785
                                50 FFFE7960   8F  D0  0070F          MOVL    #-100000, R0                                            ;
                                           02  11  00716          BRB     41$                                                     ;
                                           50  D4  00718  40$:     CLRL    R0                                                      ;
        3C  AE          50                 51 FFFCF2C0   8F  7A  0071A  41$:     EMUL    #-200000, R1, R0, DELTA_TIME                    ; 3784
                                      28   AE  9F  00724          PUSHAB  TIME1                                                   ; 3786
                                           0D  DD  00727          PUSHL   #13                                                     ;
                                      44   AE  9F  00729          PUSHAB  DELTA_TIME                                              ;
                        00000000V  00          03  FB  0072C          CALLS   #3, CONVERT_TIME                                        ;
                        00000000'  00  00000000'  00  9B  00733          MOVZBW  QUOTA1, FAODSC                                          ; 3791
                        00000000'  00  00000000'  00  9E  0073E          MOVAB   QUOTA1+1, FAODSC+4                                      ;
                                00000000'  00  DD  00749          PUSHL   RECBUF+560                                              ;
                             7E 00000000'  00  3C  0074F          MOVZWL  RECBUF+536, -(SP)                                       ;
                             7E 00000000'  00  3C  00756          MOVZWL  RECBUF+518, -(SP)                                       ;
                                00000000'  00  9F  0075D          PUSHAB  DISDSC                                                  ;
                     7E 00000000'  00          22  C1  00763          ADDL3   #34, RABPTR, -(SP)                                      ;
                                0C000000'  00  9F  0076B          PUSHAB  FAODSC                                                  ;
                        00000000G  00          06  FB  00771          CALLS   #6, SYS$FAO                                             ;
                                00000000'  00  DD  00778          PUSHL   RABPTR                                                  ;
                        00000000G  00          01  FB  0077E          CALLS   #1, SYS$PUT                                             ;
                        00000000'  00  00000000'  00  9B  00785          MOVZBW  QUOTA2, FAODSC                                          ; 3796
                        00000000'  00  00000000'  00  9E  00790          MOVAB   QUOTA2+1, FAODSC+4                                      ;
                                00000000'  00  DD  0079B          PUSHL   RECBUF+564                                              ;
                             7E 00000000'  00  3C  007A1          MOVZWL  RECBUF+538, -(SP)                                       ;
                             7E 00000000'  00  3C  007A8          MOVZWL  RECBUF+520, -(SP)                                       ;
                                00000000'  00  9F  007AF          PUSHAB  DISDSC                                                  ;
                     7E 00000000'  00          22  C1  007B5          ADDL3   #34, RABPTR, -(SP)                                      ;
                                00000000'  00  9F  007BD          PUSHAB  FAODSC                                                  ;
                        00000000G  00          06  FB  007C3          CALLS   #6, SYS$FAO                                             ;
                                00000000'  00  DD  007CA          PUSHL   RABPTR                                                  ;
                        00000000G  00          01  FB  007D0          CALLS   #1, SYS$PUT                                             ;
                        00000000'  00  00000000'  00  9B  007D7          MOVZBW  QUOTA3, FAODSC                                          ; 3801
```

```
                                ;  display_full - writes the full user display
                                ;
                                ;  Inputs:
                                ;       none
                                ;
                                ;  Outputs:
                                ;       R0 = status
                                ;
                                ;  Side effects:
                                ;       none
                                ;
                                ;--

                                BEGIN
                                LOCAL
                                    STATUS;

                                ;
                                ; Format and output the quotas
                                ;

                                STATUS = $FAO (UPLIT BYTE (%ASCII'!/Quotas:!/  Paging file quota:  !UL!/  Direct I/O limit:  !UL!/  AST limit:  !UL!/  Enqueue limit:  !UL!/  Timer queue entry limit:  !UL!/  Buffered I/O limit:  !UL!/  Buffered I/O byte limit:  !UL!/  Open file limit:  !UL!/  Subprocess limit:  !UL!/  Default CPU time:  !UL!/  Default page fault cluster:  !UL!/  Maximum AST queue limit:  !UL!/  Maximum timer queue limit:  !UL!/  Maximum buffered I/O limit:  !UL!/  Maximum buffered I/O byte limit:  !UL!/  Maximum open file limit:  !UL!/  Maximum subprocess limit:  !UL!/  Maximum CPU time:  !UL!/  Maximum page fault cluster:  !UL!/  Maximum working set limit:  !UL!/  Default working set limit:  !UL!/  Default working set extent:  !UL!/  Maximum working set extent:  !UL!/  Default page file extent:  !UL!/  Maximum page file extent:  !UL!/  Default page file end:  !UL!/  Maximum page file end:  !UL!/  Default swap file end:  !UL!/  Maximum swap file end:  !UL!/  Default swap file extent:  !UL!/  Maximum swap file extent
```

```
                                00000000'  00  9F  00985          PUSHAB  DISDSC                                                    ;
                             7E 00000000'  00  22  C1  0098B      ADDL3   #34, RABPTR, -(SP)                                      ;
                                00000000'  00  9F  00993          PUSHAB  FAODSC                                                   ;
                                00000000G  00  03  FB  00999      CALLS   #3, SYS$FAO                                              ;
                                00000000'  00  DD  009A0          PUSHL   RABPTR                                                   ;
                                00000000G  00  01  FB  009A6      CALLS   #1, SYS$PUT                                              ;
                                00000000'  00  9F  009AD          PUSHAB  RECBUF+412                                               ; 3824
                                00000000V  00  01  FB  009B3      CALLS   #1, PRINT_PRIV                                           ;
                                00000000'  00  00000000'  00  9B  009BA  MOVZBW  DEFPRIVS, FAODSC                                  ; 3826
                                00000000'  00  00000000'  00  9E  009C5  MOVAB   DEFPRIVS+1, FAODSC+4                              ;
                                00000000'  00  9F  009D0          PUSHAB  DISDSC                                                   ;
                             7E 00000000'  00  22  C1  009D6      ADDL3   #34, RABPTR, -(SP)                                      ;
                                00000000'  00  9F  009DE          PUSHAB  FAODSC                                                   ;
                                00000000G  00  03  FB  009E4      CALLS   #3, SYS$FAO                                              ;
                                00000000'  00  DD  009EB          PUSHL   RABPTR                                                   ;
                                00000000G  00  01  FB  009F1      CALLS   #1, SYS$PUT                                              ;
                                00000000'  00  9F  009F8          PUSHAB  RECBUF+420                                               ; 3827
                                00000000V  00  01  FB  009FE      CALLS   #1, PRINT_PRIV                                           ;
                                      1B 00000000'  00  E9  00A05  BLBC    RDB_EXISTS, 42$                                         ; 3833
                                00000000G  00  00  FB  00A0C      CALLS   #0, UAF$BUILD_HOLDER                                     ; 3836
                                00000000G  00  01  90  00A13      MOVB    #1, RDB_HEADER_FLAG                                      ; 3837
                                00000000G  00  9F  00A1A          PUSHAB  HOLDER                                                   ; 3838
                                00000000G  00  01  FB  00A20      CALLS   #1, UAF$WRITE_RIGHTS                                     ;
                                               04  00A27  42$:    RET                                                              ; 3842

; Routine Size:  2600 bytes,    Routine Base:  $CODE$ + 16CD
```

```
;3775          3843  1 %sbttl 'display_hours - display hourly restrictions'
;3776          3844  1 routine display_hours (format_string, hour_vector) : novalue =
;3777          3845  2 begin
;3778          3846  2
;3779          3847  2 !++
;3780          3848  2 !
;3781          3849  2 ! FUNCTIONAL DESCRIPTION:
;3782          3850  2 !
;3783          3851  2 !       Displays the hourly access for primary and secondary days
;3784          3852  2 !       for one access type.
;3785          3853  2 !
;3786          3854  2 ! INPUTS:
;3787          3855  2 !
;3788          3856  2 !       format_string: address of FAO string for display
;3789          3857  2 !       hour_vector: address of UAF record hourly vector
;3790          3858  2 !       RABPTR - RMS data structure for the file
;3791          3859  2 !
;3792          3860  2 ! IMPLICIT INPUTS:
;3793          3861  2 !
;3794          3862  2 !       none
;3795          3863  2 !
;3796          3864  2 ! OUTPUTS:
;3797          3865  2 !
;3798          3866  2 !       none
;3799          3867  2 !
;3800          3868  2 ! IMPLICIT OUTPUTS:
;3801          3869  2 !
;3802          3870  2 !       none
;3803          3871  2 !
;3804          3872  2 ! ROUTINE VALUE:
;3805          3873  2 !
;3806          3874  2 !       none
;3807          3875  2 !
;3808          3876  2 ! SIDE EFFECTS:
;3809          3877  2 !
;3810          3878  2 !       none
;3811          3879  2 !--
;3812          3880  2
;3813          3881  2 !
;3814          3882  2 ! Display strings.
;3815          3883  2 !
;3816          3884  2
;3817          3885  2 map
;3818          3886  2     hour_vector : ref bitvector;
;3819          3887  2
;3820          3888  2 literal
;3821          3889  2         yeschar         = '#';
;3822          3890  2         nochar          = '-';
;3823          3891  2
;3824          3892  2 bind
;3825          3893  2         noaccess        = cstring ('-----  No access  ------'),
;3826          3894  2         fullaccess      = cstring ('##### Full access ######');
;3827          3895  2
;3828          3896  2 local
;3829          3897  2         pri_access      : vector [24, byte],    ! Character string for primary access
;3830          3898  2         sec_access      : vector [24, byte];    ! Character string for secondary access
;3831          3899  2
```

```
;  3832          3900  2 if . (hour_vector[0])<0,24> eql 0
;  3833          3901  2 then ch$move (24, fullaccess+1, pri_access)
;  3834          3902  2 else if . (hour_vector[0])<0,24> eql %x'ffffff'
;  3835          3903  2 then ch$move (24, noaccess+1, pri_access)
;  3836          3904  2 else incr j from 0 to 23
;  3837          3905  2 do
;  3838          3906  3     begin
;  3839          3907  3     if .hour_vector[.j]
;  3840          3908  3     then pri_access[.j] = nochar
;  3841          3909  3     else pri_access[.j] = yeschar;
;  3842          3910  2     end;
;  3843          3911  2
;  3844          3912  2 if . (hour_vector[0])<24,24> eql 0
;  3845          3913  2 then ch$move (24, fullaccess+1, sec_access)
;  3846          3914  2 else if . (hour_vector[0])<24,24> eql %x'ffffff'
;  3847          3915  2 then ch$move (24, noaccess+1, sec_access)
;  3848          3916  2 else incr j from 0 to 23
;  3849          3917  2 do
;  3850          3918  3     begin
;  3851          3919  3     if .hour_vector[.j+24]
;  3852          3920  3     then sec_access[.j] = nochar
;  3853          3921  3     else sec_access[.j] = yeschar;
;  3854          3922  2     end;
;  3855          3923  2
;  3856        P 3924  2 faomac (.format_string,
;  3857        P 3925  2         24, pri_access,
;  3858        P 3926  2         24, sec_access
;  3859          3927  2 );
;  3860          3928  1 end;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                 18  0081C P.AEZ:  .BYTE   24
73  65  63  63  61  20  6F  4E  20  20  2D  2D  2D  2D  2D  0081D          .ASCII  \-----  No access  ------\                ;
                         2D  2D  2D  2D  2D  2D  20  20  73  0082C                                                          ;
                                                 18  00835 P.AFA:  .BYTE   24                                             ;
65  63  63  61  20  6C  6C  75  46  20  23  23  23  23  23  00836          .ASCII  \###### Full access ######\               ;
                         23  23  23  23  23  23  20  73  73  00845                                                          ;

                                                NOACCESS=           P.AEZ
                                                FULLACCESS=         P.AFA

                                                        .PSECT  $CODE$,NOWRT,2

                                                 01FC 00000 DISPLAY_HOURS:
                                                            .WORD   Save R2,R3,R4,R5,R6,R7,R8                           ; 3844
                                    58 00000000'  00  9E 00002          MOVAB   FULLACCESS+1, R8                        ;
                                    57 00000000'  00  9E 00009          MOVAB   FAODSC, R7                              ;
                                    5E            30  C2 00010          SUBL2   #48, SP                                 ;
                                    56      08    AC  D0 00013          MOVL    HOUR_VECTOR, R6                         ; 3900
                 00         66      18            00  ED 00017          CMPZV   #0, #24, (R6), #0                       ;
                                                  07  12 0001C          BNEQ    1$                                      ;
                       18    AE      68            18  28 0001E          MOVC3   #24, FULLACCESS+1, PRI_ACCESS           ; 3901
                                                  29  11 00023          BRB     6$                                      ;
```

```
00FFFFFF 8F            66            18                00 ED 00025 1$:     CMPZV   #0, #24, (R6), #16777215          ; 3902
                                                       08 12 0002E         BNEQ    2$                                ;
                 18    AE      E7    A8                18 28 00030         MOVC3   #24, NOACCESS+1, PRI_ACCESS       ; 3903
                                                       16 11 00036         BRB     6$                                ;
                       07            66                50 D4 00038 2$:     CLRL    J                                 ; 3904
                                  18 AE40                50 E1 0003A 3$:     BBC     J, (R6), 4$                       ; 3907
                                                       2D 90 0003E         MOVB    #45, PRI_ACCESS[J]                ; 3908
                                                       05 11 00043         BRB     5$                                ;
                                  18 AE40              23 90 00045 4$:     MOVB    #35, PRI_ACCESS[J]                ; 3909
                       EC            50                17 F3 0004A 5$:     AOBLEQ  #23, J, 3$                        ; 3904
                 50    66            18                18 EF 0004E 6$:     EXTZV   #24, #24, (R6), R0                ; 3912
                                                       06 12 00053         BNEQ    7$                                ;
                       6E            68                18 28 00055         MOVC3   #24, FULLACCESS+1, SEC_ACCESS     ; 3913
                                                       28 11 00059         BRB     12$                               ;
                    00FFFFFF   8F                      50 D1 0005B 7$:     CMPL    R0, #16777215                     ; 3914
                                                       07 12 00062         BNEQ    8$                                ;
                       6E      E7    A8                18 28 00064         MOVC3   #24, NOACCESS+1, SEC_ACCESS       ; 3915
                                                       18 11 00069         BRB     12$                               ;
                                                       50 D4 0006B 8$:     CLRL    J                                 ; 3916
                                  51          18       A0 9E 0006D 9$:     MOVAB   24(R0), R1                        ; 3919
                       06            66                51 E1 00071         BBC     R1, (R6), 10$                     ;
                                  6E40                 2D 90 00075         MOVB    #45, SEC_ACCESS[J]                ; 3920
                                                       04 11 00079         BRB     11$                               ;
                                  6E40                 23 90 0007B 10$:    MOVB    #35, SEC_ACCESS[J]                ; 3921
                       EA            50                17 F3 0007F 11$:    AOBLEQ  #23, J, 9$                        ; 3916
                                     67          04    BC 9B 00083 12$:    MOVZBW  @FORMAT_STRING, FAODSC            ; 3927
                 04    A7      04    AC                01 C1 00087         ADDL3   #1, FORMAT_STRING, FAODSC+4        ;
                                                       5E DD 0008D         PUSHL   SP                                ;
                                                       18 DD 0008F         PUSHL   #24                               ;
                                           20          AE 9F 00091         PUSHAB  PRI_ACCESS                        ;
                                                       18 DD 00094         PUSHL   #24                               ;
                                           F8          A7 9F 00096         PUSHAB  DISDSC                            ;
                 7E          08    A7                22 C1 00099         ADDL3   #34, RABPTR, -(SP)                ;
                                                       57 DD 0009E         PUSHL   R7                                ;
                    00000000G  00                      07 FB 000A0         CALLS   #7, SYS$FAO                       ;
                                           08          A7 DD 000A7         PUSHL   RABPTR                            ;
                    00000000G  00                      01 FB 000AA         CALLS   #1, SYS$PUT                       ;
                                                       04 000B1            RET                                       ; 3928

; Routine Size:  178 bytes,    Routine Base:  $CODE$ + 20F5
```

```
;  3862          3929  1 %sbttl 'convert_time - convert time value to string'
;  3863          3930  1 routine convert_time (time, length, buffer) : novalue =
;  3864          3931  2 begin
;  3865          3932  2
;  3866          3933  2 !++
;  3867          3934  2 !
;  3868          3935  2 ! FUNCTIONAL DESCRIPTION:
;  3869          3936  2 !
;  3870          3937  2 !       Converts the binary time value into a string, substituting
;  3871          3938  2 !       the string " (none)" if the value is zero.
;  3872          3939  2 !
;  3873          3940  2 ! INPUTS:
;  3874          3941  2 !
;  3875          3942  2 !       time: address of quadword time
;  3876          3943  2 !       length: length of output string
;  3877          3944  2 !       buffer: address of output buffer
;  3878          3945  2 !
;  3879          3946  2 ! IMPLICIT INPUTS:
;  3880          3947  2 !
;  3881          3948  2 !       none
;  3882          3949  2 !
;  3883          3950  2 ! OUTPUTS:
;  3884          3951  2 !
;  3885          3952  2 !       none
;  3886          3953  2 !
;  3887          3954  2 ! IMPLICIT OUTPUTS:
;  3888          3955  2 !
;  3889          3956  2 !       none
;  3890          3957  2 !
;  3891          3958  2 ! ROUTINE VALUE:
;  3892          3959  2 !
;  3893          3960  2 !       none
;  3894          3961  2 !
;  3895          3962  2 ! SIDE EFFECTS:
;  3896          3963  2 !
;  3897          3964  2 !       none
;  3898          3965  2 !--
;  3899          3966  2
;  3900          3967  2 map
;  3901          3968  2         time            : ref vector;
;  3902          3969  2
;  3903          3970  2 local
;  3904          3971  2         string_desc     : vector [2];   ! descriptor for buffer
;  3905          3972  2
;  3906          3973  2 if (.time[0] or .time[1]) eql 0
;  3907          3974  2 then
;  3908          3975  3     begin
;  3909          3976  3     ch$fill (' ', .length-6, .buffer);
;  3910          3977  3     ch$move (6, uplit byte ('(none)'), .buffer+.length-6);
;  3911          3978  3     end
;  3912          3979  3
;  3913          3980  2 else
;  3914          3981  3     begin
;  3915          3982  3     string_desc[0] = .length;
;  3916          3983  3     string_desc[1] = .buffer;
;  3917          3984  3     $asctim (timadr = .time, timbuf = string_desc);
;  3918          3985  2     end;
```

```
; 3919          3986  1 end;


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                29  65  6E  6F  6E  28  0084E P.AFB:  .ASCII  \(none)\                                ;

                                                                .EXTRN  SYS$ASCTIM

                                                                .PSECT  $CODE$,NOWRT,2

                                                    007C 00000 CONVERT_TIME:
                                                                .WORD   Save R2,R3,R4,R5,R6                             ; 3930
                                    5E              08  C2 00002        SUBL2   #8, SP                                  ;
                                    56          04  AC  D0 00005        MOVL    TIME, R6                                ; 3973
                        50          66          04  A6  C9 00009        BISL3   4(R6), (R6), R0                         ;
                                                    1C  12 0000E        BNEQ    1$                                      ;
                        50      08  AC              06  C3 00010        SUBL3   #6, LENGTH, R0                          ; 3976
                50      20          6E              00  2C 00015        MOVC5   #0, (SP), #32, R0, @BUFFER              ;
                                                0C  BC      0001A                                                       ;
                        50      0C  AC          08  AC  C1 0001C        ADDL3   LENGTH, BUFFER, R0                      ; 3977
                    FA  A0 00000000'    00              06  28 00022    MOVC3   #6, P.AFB, -6(R0)                       ;
                                                        04  0002B       RET                                             ; 3973
                                    6E          08  AC  7D 0002C 1$:    MOVQ    LENGTH, STRING_DESC                     ; 3982
                                                    7E  D4 00030        CLRL    -(SP)                                   ; 3984
                                                    56  DD 00032        PUSHL   R6                                      ;
                                                08  AE  9F 00034        PUSHAB  STRING_DESC                             ;
                                                    7E  D4 00037        CLRL    -(SP)                                   ;
                        00000000G   00              04  FB 00039        CALLS   #4, SYS$ASCTIM                          ;
                                                        04  00040       RET                                             ; 3986

; Routine Size:  65 bytes,    Routine Base:  $CODE$ + 21A7
```

```
;  3921     3987  1 %sbttl 'print_priv - print privilege bits'
;  3922     3988  1 routine print_priv (prvadr) : novalue =
;  3923     3989  2 begin
;  3924     3990  2
;  3925     3991  2 !++
;  3926     3992  2 !
;  3927     3993  2 ! FUNCTIONAL DESCRIPTION:
;  3928     3994  2 !
;  3929     3995  2 !       Routine to output the names of the privilege bits set
;  3930     3996  2 !       in the privilege vector supplied.
;  3931     3997  2 !
;  3932     3998  2 ! INPUTS:
;  3933     3999  2 !
;  3934     4000  2 !       PRVADR - Address of the privilege vector
;  3935     4001  2 !
;  3936     4002  2 ! IMPLICIT INPUTS:
;  3937     4003  2 !
;  3938     4004  2 !       PRV$AB_NAMES - table of privilege names and bit numbers
;  3939     4005  2 !
;  3940     4006  2 ! OUTPUTS:
;  3941     4007  2 !
;  3942     4008  2 !       none
;  3943     4009  2 !
;  3944     4010  2 ! IMPLICIT OUTPUTS:
;  3945     4011  2 !
;  3946     4012  2 !       none
;  3947     4013  2 !
;  3948     4014  2 ! ROUTINE VALUE:
;  3949     4015  2 !
;  3950     4016  2 !       none
;  3951     4017  2 !
;  3952     4018  2 ! SIDE EFFECTS:
;  3953     4019  2 !
;  3954     4020  2 !       none
;  3955     4021  2 !
;  3956     4022  2 !--
;  3957     4023  2
;  3958     4024  2 local
;  3959     4025  2         pointer,                        ! current location in PRV$AB_NAMES
;  3960     4026  2         prvcnt,                         ! number of names in DISBUF
;  3961     4027  2         symlen,                         ! length of bit name string
;  3962     4028  2         symmin,                         ! minimum symbol length
;  3963     4029  2         symval;                         ! value (bit number)
;  3964     4030  2
;  3965     4031  2 !
;  3966     4032  2 ! Initialize the buffer.
;  3967     4033  2 !
;  3968     4034  2
;  3969     4035  2 disbuf = ' ';                           ! insert blank at start
;  3970     4036  2 prvcnt = 0;
;  3971     4037  2 rabptr[rab$w_rsz] = 1;
;  3972     4038  2 pointer = prv$ab_names;                 ! point to symbol name table
;  3973     4039  2
;  3974     4040  2 while (symmin = . (.pointer)<0,8>) neq 0 ! pick up min symbol size
;  3975     4041  2 do
;  3976     4042  3     begin
;  3977     4043  3
```

```
; 3978     4044  3 !
; 3979     4045  3 ! Pick up the next bit name and number.  If the bit is set, insert
; 3980     4046  3 ! the bit name into the buffer.   When the buffer fills up output them
; 3981     4047  3 ! to the user.
; 3982     4048  3 !
; 3983     4049  3
; 3984     4050  3     pointer = .pointer + 1;
; 3985     4051  3     symval = . (.pointer)<0,8>;          ! get bit number
; 3986     4052  3     pointer = .pointer + 1;
; 3987     4053  3     symlen = . (.pointer)<0,8>;          ! get name string length
; 3988     4054  3     pointer = .pointer + 1;              ! point to string
; 3989     4055  3     if . (.prvadr)<.symval,1>
; 3990     4056  3     AND CH$NEQ(.SYMLEN, .POINTER, 7, UPLIT('UPGRADE'))          !**
; 3991     4057  3     AND CH$NEQ(.SYMLEN, .POINTER, 9, UPLIT('DOWNGRADE'))       !**
; 3992     4058  3     AND CH$NEQ(.SYMLEN, .POINTER, 6, UPLIT('TMPJNL'))          !**
; 3993     4059  3     AND CH$NEQ(.SYMLEN, .POINTER, 6, UPLIT('PRMJNL'))          !**
; 3994     4060  3     then
; 3995     4061  4         begin
; 3996     4062  4
; 3997     4063  4 !
; 3998     4064  4 ! Bit is set.  See if there's room in the buffer and insert it if so,
; 3999     4065  4 ! else output the buffer and start from scratch.
; 4000     4066  4 !
; 4001     4067  4
; 4002     4068  4         if .rabptr[rab$w_rsz] + .symlen geq 64
; 4003     4069  4         then
; 4004     4070  5             begin
; 4005     4071  5             $put (rab = .rabptr);
; 4006     4072  5             prvcnt = 0;
; 4007     4073  5             rabptr[rab$w_rsz] = 1;
; 4008     4074  4             end;
; 4009     4075  4
; 4010     4076  4 !
; 4011     4077  4 ! Insert a blank and append symbol name.
; 4012     4078  4 !
; 4013     4079  4
; 4014     4080  4         disbuf[.rabptr[rab$w_rsz]] = %char (' ');
; 4015     4081  4         rabptr[rab$w_rsz] = .rabptr[rab$w_rsz] + 1;
; 4016     4082  4         ch$move (.symlen, .pointer, disbuf[.rabptr[rab$w_rsz]]);
; 4017     4083  4         rabptr[rab$w_rsz] = .rabptr[rab$w_rsz] + .symlen;
; 4018     4084  4         prvcnt = .prvcnt + 1;            ! one more name in buffer
; 4019     4085  3         end;
; 4020     4086  3
; 4021     4087  3     pointer = .pointer + .symlen;        ! update table pointer over name
; 4022     4088  2     end;
; 4023     4089  2
; 4024     4090  2 !
; 4025     4091  2 ! Table used up.  If anything is in the buffer, print it.
; 4026     4092  2 !
; 4027     4093  2
; 4028     4094  2 if .prvcnt gtr 0
; 4029     4095  2 then $put (rab = .rabptr);
; 4030     4096  2
; 4031     4097  1 end;
```

```
                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                00  45  44  41  52  47  50  55  00854 P.AFC:   .ASCII  \UPGRADE\<0>
                00  00  00  45  44  41  52  47  4E  57  4F  44  0085C P.AFD:   .ASCII  \DOWNGRADE\<0><0><0>
                                00  00  4C  4E  4A  50  4D  54  00868 P.AFE:   .ASCII  \TMPJNL\<0><0>
                                00  00  4C  4E  4A  4D  52  50  00870 P.AFF:   .ASCII  \PRMJNL\<0><0>



                                                                .PSECT  $CODE$,NOWRT,2

                                                        0FFC 00000 PRINT_PRIV:
                                                                .WORD   Save R2,R3,R4,R5,P6,R7,R8,R9,R10,R11                    ; 3988
                        00000000'  00           20  D0 00002     MOVL    #32, DISBUF                                             ; 4035
                                                59  D4 00009     CLRL    PRVCNT                                                  ; 4036
                                50 00000000'    00  D0 0000B     MOVL    RABPTR, R0                                              ; 4037
                                22  A0          01  B0 00012     MOVW    #1, 34(R0)                                              ;
                                57 00000000G    00  9E 00016     MOVAB   PRV$AB_NAMES, POINTER                                   ; 4038
                                5B              67  9A 0001D 1$: MOVZBL  (POINTER), SYMMIN                                       ; 4040
                                                03  12 00020     BNEQ    2$                                                      ;
                                              0098  31 00022     BRW     6$                                                      ;
                                                57  D6 00025 2$: INCL    POINTER                                                 ; 4050
                                5A              87  9A 00027     MOVZBL  (POINTER)+, SYMVAL                                      ; 4051
                                58              87  9A 0002A     MOVZBL  (POINTER)+, SYMLEN                                      ; 4053
                        03      0x  BC          5A  E0 0002D     BBS     SYMVAL, @PRVADR, 3$                                     ; 4055
                                              0082  31 00032     BRW     5$                                                      ;
07              00              67          58  2D 00035 3$:    CMPC5   SYMLEN, (POINTER), #0, #7, P.AFC                        ; 4056
                                    00000000'   00      0003A                                                                   ;
                                                76  13 0003F     BEQL    5$                                                      ;
09              00              67          58  2D 00041        CMPC5   SYMLEN, (POINTER), #0, #9, P.AFD                        ; 4057
                                    00000000'   00      00046                                                                   ;
                                                6A  13 0004B     BEQL    5$                                                      ;
06              00              67          58  2D 0004D        CMPC5   SYMLEN, (POINTER), #0, #6, P.AFE                        ; 4058
                                    00000000'   00      00052                                                                   ;
                                                5E  13 00057     BEQL    5$                                                      ;
06              00              67          58  2D 00059        CMPC5   SYMLEN, (POINTER), #0, #6, P.AFF                        ; 4059
                                    00000000'   00      0005E                                                                   ;
                                                52  13 00065     BEQL    5$                                                      ;
                                50 00000000'    00  D0 00065     MOVL    RABPTR, R0                                              ; 4068
                                51      22      A0  3C 0006C     MOVZWL  34(R0), R1                                              ;
                                51              58  C0 00070     ADDL2   SYMLEN, R1                                              ;
                                3F              51  D1 00073     CMPL    R1, #63                                                 ;
                                                16  15 00076     BLEQ    4$                                                      ;
                                                50  DD 00078     PUSHL   R0                                                      ; 4071
                        000000006  00           01  FB 0007A     CALLS   #1, SYS$PUT                                             ;
                                                59  D4 00081     CLRL    PRVCNT                                                  ; 4072
                                50 00000000'    00  D0 00083     MOVL    RABPTR, R0                                              ; 4073
                                22  A0          01  B0 0008A     MOVW    #1, 34(R0)                                              ;
                                50 00000000'    00  D0 0008E 4$: MOVL    RABPTR, R0                                              ; 4080
                                56      22      A0  9E 00095     MOVAB   34(R0), R6                                              ;
                                50              66  3C 00099     MOVZWL  (R6), R0                                                ;
                        00000000'0040           20  90 0009C     MOVB    #32, DISBUF[R0]                                         ;
                                                66  B6 000A4     INCW    (R6)                                                    ; 4081
                                50              66  3C 000A6     MOVZWL  (R6), R0                                                ; 4082
    00000000'0040               67              58  28 000A9     MOVC3   SYMLEN, (POINTER), DISBUF[R0]                           ;
                                66              58  A0 000B2     ADDW2   SYMLEN, (R6)                                            ; 4083
                                                59  D6 000B5     INCL    PRVCNT                                                  ; 4084
```

```
                                57            58  C0 000B7 5$:     ADDL2   SYMLEN, POINTER                          ; 4087
                                            FF60  31 000BA         BRW     1$                                      ; 4040
                                              59  D5 000BD 6$:     TSTL    PRVCNT                                  ; 4094
                                              0D  15 000BF         BLEQ    7$                                      ;
                                00C00000'     00  DD 000C1         PUSHL   RABPTR                                  ; 4095
                        00000000G   00        01  FB 000C7         CALLS   #1, SYS$PUT                             ;
                                              04 000CE 7$:         RET                                             ; 4097

; Routine Size:  207 bytes,    Routine Base:  $CODE$ + 21E8
```

```
;4033     4098  1 %sbttl 'build_ini_recs - build system & default records'
;4034     4099  1 routine build_ini_recs : novalue =
;4035     4100  2 begin
;4036     4101  2
;4037     4102  2 !++
;4038     4103  2 !
;4039     4104  2 ! FUNCTIONAL DESCRIPTION:
;4040     4105  2 !
;4041     4106  2 !       Build the initial records for the creation of a new UAF file.
;4042     4107  2 !       The user default record is built in the DEFAULT_RECORD
;4043     4108  2 !       buffer and the system manager record is built in RECBUF.
;4044     4109  2 !
;4045     4110  2 ! INPUTS:
;4046     4111  2 !
;4047     4112  2 !       none
;4048     4113  2 !
;4049     4114  2 ! IMPLICIT INPUTS:
;4050     4115  2 !
;4051     4116  2 !       none
;4052     4117  2 !
;4053     4118  2 ! OUTPUTS:
;4054     4119  2 !
;4055     4120  2 !       none
;4056     4121  2 !
;4057     4122  2 ! IMPLICIT OUTPUTS:
;4058     4123  2 !
;4059     4124  2 !       default record is built in DEFAULT_RECORD
;4060     4125  2 !       system record is built in RECBUF
;4061     4126  2 !
;4062     4127  2 ! ROUTINE VALUE:
;4063     4128  2 !
;4064     4129  2 !       none
;4065     4130  2 !
;4066     4131  2 ! SIDE EFFECTS:
;4067     4132  2 !
;4068     4133  2 !       none
;4069     4134  2 !--
;4070     4135  2
;4071     4136  2 local
;4072     4137  2         user_desc       : $bblock [8];
;4073     4138  2
;4074     4139  2 ch$fill (0, uaf$c_fixed, default_record);
;4075     4140  2 default_record[uaf$b_version] = uaf$c_version1;
;4076     4141  2 default_record[uaf$b_rtype] = uaf$c_user_id;
;4077     4142  2 !
;4078     4143  2 ! username is blank filled
;4079     4144  2 !
;4080     4145  2
;4081     4146  2 ch$copy (.defuser<0,8>, defuser+1, %char (' '),
;4082     4147  2          uaf$s_username, default_record[uaf$t_username]);
;4083     4148  2
;4084     4149  2 !
;4085     4150  2 ! account name is blank filled
;4086     4151  2 !
;4087     4152  2
;4088     4153  2 ch$copy (.defact<0,8>, defact+1, %char (' '),
;4089     4154  2          uaf$s_account, default_record[uaf$t_account]);
```

```
;4090        4155  2
;4091        4156  2 !
;4092        4157  2 ! quadword privilege mask
;4093        4158  2 !
;4094        4159  2
;4095        4160  2 ch$move (8, defpriv, default_record[uaf$q_priv]);
;4096        4161  2 ch$move (8, defpriv, default_record[uaf$q_def_priv]);
;4097        4162  2
;4098        4163  2 !
;4099        4164  2 ! directory name is counted string
;4100        4165  2 !
;4101        4166  2
;4102        4167  2 ch$copy (.defdir<0,8> + 1, defdir, %char (' '),
;4103        4168  2          uaf$s_defdir, default_record[uaf$t_defdir]);
;4104        4169  2
;4105        4170  2 !
;4106        4171  2 ! device name is counted string
;4107        4172  2 !
;4108        4173  2
;4109        4174  2 ch$copy (.defdev<0,8> + 1, defdev, %char (' '),
;4110        4175  2          uaf$s_defdev, default_record[uaf$t_defdev]);
;4111        4176  2
;4112        4177  2 !
;4113        4178  2 ! CLI name is counted string
;4114        4179  2 !
;4115        4180  2
;4116        4181  2 ch$copy (.defcli<0,8> + 1, defcli, %char (' '),
;4117        4182  2          uaf$s_defcli, default_record[uaf$t_defcli]);
;4118        4183  2
;4119        4184  2 !
;4120        4185  2 ! owner name is counted string
;4121        4186  2 !
;4122        4187  2
;4123        4188  2 ch$copy (.defowner<0,8> + 1, defowner, %char (' '),
;4124        4189  2          uaf$s_owner, default_record[uaf$t_owner]);
;4125        4190  2
;4126        4191  2 !
;4127        4192  2 ! login command file name is counted string
;4128        4193  2 !
;4129        4194  2
;4130        4195  2 ch$copy (.deflgicmd<0,8> + 1, deflgicmd, %char (' '),
;4131        4196  2          uaf$s_lgicmd, default_record[uaf$t_lgicmd]);
;4132        4197  2
;4133        4198  2 !
;4134        4199  2 ! fill in default CLI tables
;4135        4200  2 !
;4136        4201  2 ch$copy (.defclitabl<0,8> + 1, defclitabl, %char (' '),
;4137        4202  2          uaf$s_clitables, default_record[uaf$t_clitables]);
;4138        4203  2
;4139        4204  2 ch$move (8, defpwdlife, default_record[uaf$q_pwd_lifetime]);
;4140        4205  2 default_record[uaf$b_pwd_length] = defpwdlength;
;4141        4206  2 default_record[uaf$w_grp] = defgrp;
;4142        4207  2 default_record[uaf$w_mem] = defmem;
;4143        4208  2 default_record[uaf$w_biolm] = defbiolm;
;4144        4209  2 default_record[uaf$l_bytlm] = defbytlm;
;4145        4210  2 default_record[uaf$w_diolm] = defdiolm;
;4146        4211  2 default_record[uaf$w_fillm] = deffillm;
```

```
;  4147          4212  2 default_record[uaf$l_flags] = defflags;
;  4148          4213  2 default_record[uaf$w_tqcnt] = deftqcnt;
;  4149          4214  2 default_record[uaf$w_prccnt] = defprccnt;
;  4150          4215  2 default_record[uaf$l_wsquota] = defwsquota;
;  4151          4216  2 default_record[uaf$l_wsextent] = defwsextent;
;  4152          4217  2 default_record[uaf$l_dfwscnt] = defdfwscnt;
;  4153          4218  2 default_record[uaf$l_cputim] = defcputim;
;  4154          4219  2 default_record[uaf$w_astlm] = defastlm;
;  4155          4220  2 default_record[uaf$l_pgflquota] = defpgflquota;
;  4156          4221  2 default_record[uaf$w_enqlm] = defenqlm;
;  4157          4222  2 default_record[uaf$l_pbytlm] = defpbytlm;
;  4158          4223  2 default_record[uaf$w_shrfillm] = defshrfillm;
;  4159          4224  2 default_record[uaf$b_pri] = defpri;
;  4160          4225  2 default_record[uaf$b_quepri] = defquepri;
;  4161          4226  2 default_record[uaf$w_maxjobs] = defmaxjobs;
;  4162          4227  2 default_record[uaf$w_maxdetach] = defmaxdetach;
;  4163          4228  2 default_record[uaf$l_jtquota] = defjtquota;
;  4164          4229  2 default_record[uaf$w_maxacctjobs] = defmaxacctjobs;
;  4165          4230  2 default_record[uaf$b_primedays] = defprimedays;
;  4166          4231  2 default_record[uaf$b_network_access_p] = defhours;
;  4167          4232  2 default_record[uaf$b_network_access_s] = defhours;
;  4168          4233  2 default_record[uaf$b_batch_access_p] = defhours;
;  4169          4234  2 default_record[uaf$b_batch_access_s] = defhours;
;  4170          4235  2 default_record[uaf$b_local_access_p] = defhours;
;  4171          4236  2 default_record[uaf$b_local_access_s] = defhours;
;  4172          4237  2 default_record[uaf$b_dialup_access_p] = defhours;
;  4173          4238  2 default_record[uaf$b_dialup_access_s] = defhours;
;  4174          4239  2 default_record[uaf$b_remote_access_p] = defhours;
;  4175          4240  2 default_record[uaf$b_remote_access_s] = defhours;
;  4176          4241  2
;  4177          4242  2 ch$fill (0, uaf$c_fixed, recbuf);
;  4178          4243  2 recbuf[uaf$b_version] = uaf$c_version1;
;  4179          4244  2 recbuf[uaf$b_rtype] = uaf$c_user_id;
;  4180          4245  2 ch$copy (.sysuser<0,8>, sysuser+1, %char (' '),
;  4181          4246  2           uaf$s_username, recbuf[uaf$t_username]);
;  4182          4247  2 ch$copy (.sysact<0,8>, sysact+1, %char (' '),
;  4183          4248  2           uaf$s_account, recbuf[uaf$t_account]);
;  4184          4249  2 ch$move (8, syspriv, recbuf[uaf$q_priv]);
;  4185          4250  2 ch$move (8, syspriv, recbuf[uaf$q_def_priv]);
;  4186          4251  2 ch$copy (.sysdir<0,8> + 1, sysdir, %char (' '),
;  4187          4252  2           uaf$s_defdir, recbuf[uaf$t_defdir]);
;  4188          4253  2 ch$copy (.sysdev<0,8> + 1, sysdev, %char (' '),
;  4189          4254  2           uaf$s_defdev, recbuf[uaf$t_defdev]);
;  4190          4255  2 ch$copy (.sysCLI<0,8> + 1, syscli, %char (' '),
;  4191          4256  2           uaf$s_defcli, recbuf[uaf$t_defcli]);
;  4192          4257  2 ch$copy (.sysowner<0,8> + 1, sysowner, %char (' '),
;  4193          4258  2           uaf$s_owner, recbuf[uaf$t_owner]);
;  4194          4259  2 ch$copy (.syslgicmd<0,8> + 1, deflgicmd, %char (' '),
;  4195          4260  2           uaf$s_lgicmd, recbuf[uaf$t_lgicmd]);
;  4196          4261  2 ch$copy (.sysCLITABL<0,8> + 1, sysCLITABL, %char (' '),
;  4197          4262  2           uaf$s_clitables, recbuf[uaf$t_clitables]);
;  4198          4263  2
;  4199          4264  2 pwddsc[dsc$w_length] = .syspass<0,8>;
;  4200          4265  2 pwddsc[dsc$a_pointer] = syspass+1;
;  4201          4266  2 user_desc[dsc$w_length] = .sysuser<0,8>;
;  4202          4267  2 user_desc[dsc$a_pointer] = sysuser+1;
;  4203          4268  2 $gettim (timadr = time_buf);                    ! Obtain a 16 bit salt
```

```
;  4204          4269  2 recbuf[uaf$w_salt] = .time_buf<3*8,16>;
;  4205          4270  2 recbuf[uaf$b_encrypt] = encrypt;
;  4206          4271  2 lgi$hpwd (rec_encrypt_dsc, pwddsc, .recbuf[uaf$b_encrypt],
;  4207          4272  2           .recbuf[uaf$w_salt], user_desc);
;  4208          4273  2
;  4209          4274  2 ch$move (8, syspwdlife, recbuf[uaf$q_pwd_lifetime]);
;  4210          4275  2 recbuf[uaf$b_pwd_length] = syspwdlength;
;  4211          4276  2 recbuf[uaf$w_grp] = sysgrp;
;  4212          4277  2 recbuf[uaf$w_mem] = sysmem;
;  4213          4278  2 recbuf[uaf$w_biolm] = sysbiolm;
;  4214          4279  2 recbuf[uaf$l_bytlm] = sysbytlm;
;  4215          4280  2 recbuf[uaf$w_diolm] = sysdiolm;
;  4216          4281  2 recbuf[uaf$w_fillm] = sysfillm;
;  4217          4282  2 recbuf[uaf$l_flags] = sysflags;
;  4218          4283  2 recbuf[uaf$w_tqcnt] = systqcnt;
;  4219          4284  2 recbuf[uaf$w_prccnt] = sysprccnt;
;  4220          4285  2 recbuf[uaf$l_wsquota] = syswsquota;
;  4221          4286  2 recbuf[uaf$l_wsextent] = syswsextent;
;  4222          4287  2 recbuf[uaf$l_dfwscnt] = sysdfwscnt;
;  4223          4288  2 recbuf[uaf$l_cputim] = syscputim;
;  4224          4289  2 recbuf[uaf$w_astlm] = sysastlm;
;  4225          4290  2 recbuf[uaf$l_pgflquota] = syspgflquota;
;  4226          4291  2 recbuf[uaf$w_enqlm] = sysenqlm;
;  4227          4292  2 recbuf[uaf$l_pbytlm] = syspbytlm;
;  4228          4293  2 recbuf[uaf$w_shrfillm] = sysshrfillm;
;  4229          4294  2 recbuf[uaf$b_pri] = syspri;
;  4230          4295  2 default_record[uaf$b_quepri] = sysquepri;
;  4231          4296  2 recbuf[uaf$w_maxdetach] = sysmaxdetach;
;  4232          4297  2 recbuf[uaf$l_jtquota] = sysjtquota;
;  4233          4298  2 recbuf[uaf$w_maxjobs] = sysmaxjobs;
;  4234          4299  2 recbuf[uaf$w_maxdetach] = sysmaxdetach;
;  4235          4300  2 recbuf[uaf$w_maxacctjobs] = sysmaxacctjobs;
;  4236          4301  2 recbuf[uaf$b_primedays] = sysprimedays;
;  4237          4302  2 recbuf[uaf$b_network_access_p] = defhours;
;  4238          4303  2 recbuf[uaf$b_network_access_s] = defhours;
;  4239          4304  2 recbuf[uaf$b_batch_access_p] = defhours;
;  4240          4305  2 recbuf[uaf$b_batch_access_s] = defhours;
;  4241          4306  2 recbuf[uaf$b_local_access_p] = defhours;
;  4242          4307  2 recbuf[uaf$b_local_access_s] = defhours;
;  4243          4308  2 recbuf[uaf$b_dialup_access_p] = defhours;
;  4244          4309  2 recbuf[uaf$b_dialup_access_s] = defhours;
;  4245          4310  2 recbuf[uaf$b_remote_access_p] = defhours;
;  4246          4311  2 recbuf[uaf$b_remote_access_s] = defhours;
;  4247          4312  1 end;


                                        03FC 00000 BUILD_INI_RECS:
                                                   .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9                    ; 4099
                                59 00000000'  00  9E 00002 MOVAB   DEFACT+1, R9                            ;
                                58 00000000'  00  9E 00009 MOVAB   DEFAULT_RECORD, R8                      ;
                                57 00000000'  00  9E 00010 MOVAB   RECBUF, R7                              ;
                                5E            08  C2 00017 SUBL2   #8, SP                                 ;
    0284    8F          00      6E            00  2C 0001A MOVC5   #0, (SP), #0, #644, DEFAULT_RECORD     ; 4139
                                                68     00021                                               ;
```

```
                                                68      0101    8F   B0 00022          MOVW    #257, DEFAULT_RECORD                                    ; 4141
                                                50        E8    A9   9A 00027          MOVZBL  DEFUSER, R0                                             ; 4146
            20              20          E9      A9              50   2C 0002B          MOVC5   R0, DEFUSER+1, #32, #32, DEFAULT_RECORD+4               ; 4147
                                                          04    A8      00031
                                                50        FF    A9   9A 00033          MOVZBL  DEFACT, R0                                              ; 4153
            20              20                  69              50   2C 00037          MOVC5   R0, DEFACT+1, #32, #32, DEFAULT_RECORD+52               ; 4154
                                                          34    A8      0003C
                    019C    C8          10      A9              08   28 0003E          MOVC3   #8, DEFPRIV, DEFAULT_RECORD+412                         ; 4160
                    01A4    C8          10      A9              08   28 00045          MOVC3   #8, DEFPRIV, DEFAULT_RECORD+420                         ; 4161
                                                50        06    A9   9A 0004C          MOVZBL  DEFDIR, R0                                              ; 4167
                                                                50   D6 00050          INCL    R0
        0040    8F              20          06      A9              50   2C 00052          MOVC5   R0, DEFDIR, #32, #64, DEFAULT_RECORD+148                ; 4168
                                                        0094    C8      0005A
                                                50        0D    A9   9A 0005D          MOVZBL  DEFDEV, R0                                              ; 4174
                                                                50   D6 00061          INCL    R0
            20              20          0D      A9              50   2C 00063          MOVC5   R0, DEFDEV, #32, #32, DEFAULT_RECORD+116                ; 4175
                                                          74    A8      00069
                                                50              69   9A 0006B          MOVZBL  DEFCLI, R0                                              ; 4181
                                                                50   D6 0006E          INCL    R0
            20              20                  69              50   2C 00070          MOVC5   R0, DEFCLI, #32, #32, DEFAULT_RECORD+276                ; 4182
                                                        0114    C8      00075
                                                50        04    A9   9A 00078          MOVZBL  DEFOWNER, R0                                            ; 4188
                                                                50   D6 0007C          INCL    R0
            20              20          04      A9              50   2C 0007E          MOVC5   R0, DEFOWNER, #32, #32, DEFAULT_RECORD+84               ; 4189
                                                          54    A8      00084
                                                50        05    A9   9A 00086          MOVZBL  DEFLGICMD, R0                                           ; 4195
                                                                50   D6 0008A          INCL    R0
        0040    8F              20          05      A9              50   2C 0008C          MOVC5   R0, DEFLGICMD, #32, #64, DEFAULT_RECORD+212             ; 4196
                                                        00D4    C8      00094
                                                50        F5    A9   9A 00097          MOVZBL  DEFCLITABL, R0                                          ; 4201
                                                                50   D6 0009B          INCL    R0
            20              20          F5      A9              50   2C 0009D          MOVC5   R0, DEFCLITABL, #32, #32, -                             ; 4202
                                                        0134    C8      000A3                  DEFAULT_RECORD+308
                    0174    C8          18      A9              08   28 000A6          MOVC3   #8, DEFPWDLIFE, DEFAULT_RECORD+372                      ; 4204
                                            016A    C8              06   90 000AD          MOVB    #6, DEFAULT_RECORD+362                          ; 4205
                                              24    A8 00800080   8F   D0 000B2          MOVL    #8388736, DEFAULT_RECORD+36                     ; 4207
                                            0230    C8      1000  8F   3C 000BA          MOVZWL  #4096, DEFAULT_RECORD+560                       ; 4209
                                            020E    C8 00060006   8F   D0 000C1          MOVL    #393222, DEFAULT_RECORD+526                     ; 4208
                                                        01D4    C8   D4 000CA          CLRL    DEFAULT_RECORD+468                              ; 4212
                                            020C    C8            02   B0 000CE          MOVW    #2, DEFAULT_RECORD+524                          ; 4214
                                            021C    C8      C8    8F   9A 000D3          MOVZBL  #200, DEFAULT_RECORD+540                        ; 4215
                                            0224    C8      01F4  8F   3C 000D9          MOVZWL  #500, DEFAULT_RECORD+548                        ; 4216
                                            0220    C8        96  8F   9A 000E0          MOVZBL  #150, DEFAULT_RECORD+544                        ; 4217
                                                        022C    C8   D4 000E6          CLRL    DEFAULT_RECORD+556                              ; 4218
                                            0212    C8 000A000A   8F   D0 000EA          MOVL    #655370, DEFAULT_RECORD+530                     ; 4213
                                            0228    C8      2710  8F   3C 000F3          MOVZWL  #10000, DEFAULT_RECORD+552                      ; 4220
                                            0216    C8 0014000A   8F   D0 000FA          MOVL    #1310730, DEFAULT_RECORD+534                    ; 4221
                                                        0234    C8   D4 00103          CLRL    DEFAULT_RECORD+564                              ; 4222
                                                        021A    C8   B4 00107          CLRW    DEFAULT_RECORD+538                              ; 4223
                                            0204    C8      0404  8F   3C 0010B          MOVZWL  #1028, DEFAULT_RECORD+516                       ; 4224
                                            0238    C8      0400  8F   3C 00112          MOVZWL  #1024, DEFAULT_RECORD+568                       ; 4228
                                                        0208    C8   D4 00119          CLRL    DEFAULT_RECORD+520                              ; 4229
                                            0202    C8        60  8F   90 0011D          MOVB    #96, DEFAULT_RECORD+514                         ; 4230
    01D8    C8              18                  00              00   F0 00123          INSV    #0, #0, #24, DEFAULT_RECORD+472                 ; 4231
    01DB    C8              18                  00              00   F0 0012A          INSV    #0, #0, #24, DEFAULT_RECORD+475                 ; 4232
    01DE    C8              18                  00              00   F0 00131          INSV    #0, #0, #24, DEFAULT_RECORD+478                 ; 4233
    01E1    C8              18                  00              00   F0 00138          INSV    #0, #0, #24, DEFAULT_RECORD+481                 ; 4234
```

```
                                                
```

```
                                 24   A7 00010004    8F   D0 0024A        MOVL    #65540, RECBUF+36                       ; 4277
                               0230   C7     5000    8F   3C 00252        MOVZWL  #20480, RECBUF+560                      ; 4279
                               020E   C7 000C000C    8F   D0 00259        MOVL    #786444, RECBUF+526                     ; 4278
                                           01D4    C7   D4 00262        CLRL    RECBUF+468                              ; 4282
                               0212   C7 00140014    8F   D0 00266        MOVL    #1310740, RECBUF+530                    ; 4283
                               020A   C7 000A0000    8F   D0 0026F        MOVL    #655360, RECBUF+522                     ; 4296
                               021C   C7     015E    8F   3C 00278        MOVZWL  #350, RECBUF+540                        ; 4285
                               0224   C7     0400    8F   3C 0027F        MOVZWL  #1024, RECBUF+548                       ; 4286
                               0220   C7       96    8F   9A 00286        MOVZBL  #150, RECBUF+544                        ; 4287
                                           022C    C7   D4 0028C        CLRL    RECBUF+556                              ; 4288
                               0228   C7     2710    8F   3C 00290        MOVZWL  #10000, RECBUF+552                      ; 4290
                               0216   C7 00140014    8F   D0 00297        MOVL    #1310740, RECBUF+534                    ; 4291
                                           0234    C7   D4 002A0        CLRL    RECBUF+564                              ; 4292
                                           021A    C7   B4 002A4        CLRW    RECBUF+538                              ; 4293
                               0204   C7             04   90 002A8        MOVB    #4, RECBUF+516                          ; 4294
                               0205   C8             04   90 002AD        MOVB    #4, DEFAULT_RECORD+517                  ; 4295
                               0238   C7     0400    8F   3C 002B2        MOVZWL  #1024, RECBUF+568                       ; 4297
                                           020A    C7   B4 002B9        CLRW    RECBUF+522                              ; 4299
                                           0206    C7   D4 002BD        CLRL    RECBUF+518                              ; 4298
                               0202   C7       60    8F   90 002C1        MOVB    #96, RECBUF+514                         ; 4301
    01D8   C7          18             00             00   F0 002C7        INSV    #0, #0, #24, RECBUF+472                 ; 4302
    01DB   C7          18             00             00   F0 002CE        INSV    #0, #0, #24, RECBUF+475                 ; 4303
    01DE   C7          18             00             00   F0 002D5        INSV    #0, #0, #24, RECBUF+478                 ; 4304
    01E1   C7          18             00             00   F0 002DC        INSV    #0, #0, #24, RECBUF+481                 ; 4305
    01E4   C7          18             00             00   F0 002E3        INSV    #0, #0, #24, RECBUF+484                 ; 4306
    01E7   C7          18             00             00   F0 002EA        INSV    #0, #0, #24, RECBUF+487                 ; 4307
    01EA   C7          18             00             00   F0 002F1        INSV    #0, #0, #24, RECBUF+490                 ; 4308
    01ED   C7          18             00             00   F0 002F8        INSV    #0, #0, #24, RECBUF+493                 ; 4309
    01F0   C7          18             00             00   F0 002FF        INSV    #0, #0, #24, RECBUF+496                 ; 4310
    01F3   C7          18             00             00   F0 00306        INSV    #0, #0, #24, RECBUF+499                 ; 4311
                                                           04 0030D        RET                                             ; 4312

; Routine Size:  782 bytes,    Routine Base:  $CODE$ + 22B7
```

```
;  4249          4313  1 %sbttl 'get_user_record - get username and lookup record'
;  4250          4314  1 routine get_user_record (lock_record, permanent_ok) =
;  4251          4315  2 begin
;  4252          4316  2
;  4253          4317  2 !++
;  4254          4318  2 !
;  4255          4319  2 ! FUNCTIONAL DESCRIPTION:
;  4256          4320  2 !
;  4257          4321  2 !       This routine pulls the next token out of the command
;  4258          4322  2 !       buffer, assuming it is the username, and looks up the
;  4259          4323  2 !       UAF record for that name.  If the record is found,
;  4260          4324  2 !       it is loaded into RECBUF (by routine LOCATE_USER).
;  4261          4325  2 !
;  4262          4326  2 ! INPUTS:
;  4263          4327  2 !
;  4264          4328  2 !       LOCK_RECORD - specifies that the GET shall lock the record
;  4265          4329  2 !       PERMANENT_OK - specifies that the DEFAULT and SYSTEM records are allowed
;  4266          4330  2 !
;  4267          4331  2 ! IMPLICIT INPUTS:
;  4268          4332  2 !
;  4269          4333  2 !       TOKENPTR - address of delimiter following last token processed,
;  4270          4334  2 !                  which was the command name.
;  4271          4335  2 !       TOKENLEN - global variable to contain length of current token
;  4272          4336  2 !
;  4273          4337  2 ! OUTPUTS:
;  4274          4338  2 !
;  4275          4339  2 !       none
;  4276          4340  2 !
;  4277          4341  2 ! IMPLICIT OUTPUTS:
;  4278          4342  2 !
;  4279          4343  2 !       none
;  4280          4344  2 !
;  4281          4345  2 ! ROUTINE VALUE:
;  4282          4346  2 !
;  4283          4347  2 !       true -> user record found
;  4284          4348  2 !       false -> user record not found
;  4285          4349  2 !
;  4286          4350  2 ! SIDE EFFECTS:
;  4287          4351  2 !
;  4288          4352  2 !       none
;  4289          4353  2 !--
;  4290          4354  2
;  4291          4355  2 builtin
;  4292          4356  2         nullparameter;
;  4293          4357  2
;  4294          4358  2 !
;  4295          4359  2 ! Is this the second phase of a RENAME?
;  4296          4360  2 !
;  4297          4361  2 if .rename_ph2
;  4298          4362  2 then
;  4299          4363  3     begin
;  4300          4364  3     if .netuaf_exists
;  4301          4365  3     then adjust_proxy (update_records);
;  4302          4366  3     ch$move (.olduserlen, oldusername, .tokenptr);
;  4303          4367  3     tokenlen = .olduserlen;
;  4304          4368  3     rename_ph2 = false;
;  4305          4369  3     end
```

```
;  4306          4370  3 !
;  4307          4371  3 ! Not the second phase of a RENAME.  Get first token, and if this
;  4308          4372  3 ! is the first phase of a RENAME (COPY phase), save token for
;  4309          4373  3 ! the second phase (REMOVE phase).
;  4310          4374  3 !
;  4311          4375  2 else
;  4312          4376  3     begin
;  4313          4377  3 !
;  4314          4378  3 ! Get token
;  4315          4379  3 !
;  4316          4380  3         if not cli$present (sd_token1)
;  4317          4381  3         or not cli$get_value (sd_token1,tokendsc)
;  4318          4382  3         or .tokenlen eql 0
;  4319          4383  3         then return LIB$SIGNAL(UAF$_NOUSERNAME);
;  4320          4384  3 !
;  4321          4385  3 ! If the third argument is present, this is the first phase of a
;  4322          4386  3 ! RENAME,  so save the token for the next call
;  4323          4387  3 !
;  4324          4388  3         if not nullparameter (3)
;  4325          4389  3         then
;  4326          4390  4             begin
;  4327          4391  4             ch$copy (.tokenlen, .tokenptr, ' ', uaf$s_username, oldusername);
;  4328          4392  4             olduserlen = .tokenlen;
;  4329          4393  4             rename_ph2 = true;
;  4330          4394  3             end;
;  4331          4395  2         end;
;  4332          4396  2
;  4333          4397  2 if not .permanent_ok
;  4334          4398  2 then
;  4335          4399  3     begin
;  4336          4400  3     if ch$eql (.defuser<0,8>, defuser+1, .tokenlen, .tokenptr, ' ')
;  4337          4401  3     then
;  4338          4402  3         if nullparameter (3)
;  4339          4403  3         then
;  4340          4404  3             return LIB$SIGNAL(UAF$_REMDEF)
;  4341          4405  3         else
;  4342          4406  3             return LIB$SIGNAL(UAF$_RENDEF);
;  4343          4407  3
;  4344          4408  3     if ch$eql (.sysuser<0,8>, sysuser+1, .tokenlen, .tokenptr, ' ')
;  4345          4409  3     then
;  4346          4410  4         begin
;  4347          4411  4         if nullparameter (3)
;  4348          4412  4         then
;  4349          4413  4             return LIB$SIGNAL(UAF$_REMSYS)
;  4350          4414  4         else
;  4351          4415  4             return LIB$SIGNAL(UAF$_RENSYS);
;  4352          4416  3         end;                                                          ;
;  4353          4417  2     end;
;  4354          4418  2
;  4355          4419  2 if locate_user (.tokenlen, .tokenptr, .lock_record)
;  4356          4420  2 then return true;
;  4357          4421  2
;  4358          4422  2 if .rmserr eql rms$_rnf
;  4359          4423  2 then
;  4360          4424  2     LIB$SIGNAL(UAF$_BADUSR, 2, .tokenlen, .tokenptr)
;  4361          4425  2 else
;  4362          4426  2     LIB$SIGNAL(UAF$_GETERR, 0, .rmserr);
```

```
; 4363          4427  2 false
; 4364          4428  1 end;


                                        07FC 00000 GET_USER_RECORD:
                                                   .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10                ; 4314
                                5A 00000000G    00  9E 00002           MOVAB   LIB$SIGNAL, R10
                                59 00000000'    00  9E 00009           MOVAB   SD_TOKEN1, R9
                                58 00000000'    00  9E 00010           MOVAB   RENAME_PH2, R8
                                57 00000000'    00  9E 00017           MOVAB   TOKENLEN, R7
                                1F              68  E9 0001E           BLBC    RENAME_PH2, 2$                 ; 4361
                                07        10    A7  E9 00021           BLBC    NETUAF_EXISTS, 1$              ; 4364
                                          7E    D4 00025               CLRL    -(SP)                          ; 4365
                             E796       CF      01  FB 00027           CALLS   #1, ADJUST_PROXY
                                56        04    A8  D0 0002C 1$:       MOVL    OLDUSERLEN, R6                 ; 4366
                                50        04    A7  D0 00030           MOVL    TOKENPTR, R0
                          60        08    A8      56  28 00034           MOVC3   R6, OLDUSERNAME, (R0)
                                67              56  B0 00039           MOVW    R6, TOKENLEN                   ; 4367
                                                68  94 0003C           CLRB    RENAME_PH2                     ; 4368
                                                45  11 0003E           BRB     5$                             ; 4361
                                                59  DD 00040 2$:       PUSHL   R9                             ; 4380
                                00000000G       00  FB 00042           CALLS   #1, CLI$PRESENT
                                                12  50 E9 00049        BLBC    R0, 3$
                                                    57 DD 0004C        PUSHL   R7                             ; 4381
                                                    59 DD 0004E        PUSHL   R9
                                00000000G       00  FB 00050           CALLS   #2, CLI$GET_VALUE
                                                04  50 E9 00057        BLBC    R0, 3$
                                                    67 B5 0005A        TSTW    TOKENLEN                       ; 4382
                                                    08 12 0005C        BNEQ    4$
                                      00000000G     8F DD 0005E 3$:    PUSHL   #UAF$_NOUSERNAME               ; 4383
                                                    7B 11 00064        BRB     11$
                                   03               6C 91 00066 4$:    CMPB    (AP), #3                       ; 4388
                                                    1A 1F 00069        BLSSU   5$
                                         0C         AC D5 0006B        TSTL    12(AP)
                                                    15 13 0006E        BEQL    5$
                                56                  67 3C 00070        MOVZWL  TOKENLEN, R6                   ; 4391
                                50        04        A7 D0 00073        MOVL    TOKENPTR, R0
                20        20      60              56 2C 00077        MOVC5   R6, (R0), #32, #32, OLDUSERNAME
                                          08        A8     0007C
                                04        A8      56 D0 0007E        MOVL    R6, OLDUSERLEN                 ; 4392
                                68              01 90 00082        MOVB    #1, RENAME_PH2                 ; 4393
                                5C        08    AC E8 00085 5$:      BLBS    PERMANENT_OK, 12$              ; 4397
                                51      0118    C9 9A 00089        MOVZBL  DEFUSER, R1                    ; 4400
                                50        04    A7 D0 0008E        MOVL    TOKENPTR, R0
                67      20      0119    C9      51 2D 00092        CMPC5   R1, DEFUSER+1, #32, TOKENLEN, (R0)
                                                60     00099
                                                1A 12 0009A        BNEQ    8$
                                   03           6C 91 0009C        CMPB    (AP), #3                       ; 4402
                                                05 1F 0009F        BLSSU   6$
                                         0C     AC D5 000A1        TSTL    12(AP)
                                                08 12 000A4        BNEQ    7$
                                      00000000G 8F DD 000A6 6$:    PUSHL   #UAF$_REMDEF                   ; 4404
                                                33 11 000AC        BRB     11$
                                      00000000G 8F DD 000AE 7$:    PUSHL   #UAF$_RENDEF                   ; 4406
```

```
                                                2B  11 000B4          BRB     11$                                                     ;
                                          51    C9  9A 000B6 8$:     MOVZBL  SYSUSER, R1                                             ; 4408
                                          50    04  A7  D0 000BB      MOVL    TOKENPTR, R0                                            ;
                67              20  0151   C9        51  2D 000BF      CMPC5   R1, SYSUSER+1, #32, TOKENLEN, (R0)                      ;
                                                    60     000C6                                                                      ;
                                                1C  12 000C7          BNEQ    12$                                                     ;
                                          03        6C  91 000C9      CMPB    (AP), #3                                                ; 4411
                                                    05  1F 000CC      BLSSU   9$                                                      ;
                                             0C     AC  D5 000CE      TSTL    12(AP)                                                  ;
                                                    08  12 000D1      BNEQ    10$                                                     ;
                                       00000000G    8F  DD 000D3 9$:     PUSHL   #UAF$_REMSYS                                        ; 4413
                                                    06  11 000D9      BRB     11$                                                     ;
                                       00000000G    8F  DD 000DB 10$:    PUSHL   #UAF$_RENSYS                                        ; 4415
                                          6A            01  FB 000E1 11$:    CALLS   #1, LIB$SIGNAL                                  ;
                                                        04  000E4      RET                                                             ;
                                             04     AC  DD 000E5 12$:    PUSHL   LOCK_RECORD                                         ; 4419
                                             04     A7  DD 000E8      PUSHL   TOKENPTR                                                ;
                                          7E        67  3C 000EB      MOVZWL  TOKENLEN, -(SP)                                         ;
                        00000000V         00        03  FB 000EE      CALLS   #3, LOCATE_USER                                         ;
                                                    04  50  E9 000F5      BLBC    R0, 13$                                             ;
                                                    50  01  D0 000F8      MOVL    #1, R0                                              ; 4420
                                                        04  000FB      RET                                                             ;
                                                    50      18  A7  D0 000FC 13$:    MOVL    RMSERR, R0                              ; 4422
                        000182B2                    8F      50  D1 00100      CMPL    R0, #98994                                      ;
                                                            13  12 00107      BNEQ    14$                                             ;
                                                        04  A7  DD 00109      PUSHL   TOKENPTR                                        ; 4424
                                                    7E      67  3C 0010C      MOVZWL  TOKENLEN, -(SP)                                 ;
                                                            02  DD 0010F      PUSHL   #2                                              ;
                                                00000000G   8F  DD 00111      PUSHL   #UAF$_BADUSR                                    ;
                                                    6A          04  FB 00117      CALLS   #4, LIB$SIGNAL                              ;
                                                            0D  11 0011A      BRB     15$                                             ;
                                                            50  DD 0011C 14$:     PUSHL   R0                                      ; 4426
                                                            7E  D4 0011E      CLRL    -(SP)                                           ;
                                                00000000G   8F  DD 00120      PUSHL   #UAF$_GETERR                                    ;
                                                    6A          03  FB 00126      CALLS   #3, LIB$SIGNAL                              ;
                                                            50  D4 00129 15$:     CLRL    R0                                      ; 4428
                                                                04  0012B      RET                                                     ;

; Routine Size:  300 bytes,    Routine Base:  $CODE$ + 25C5
```

```
;  4366     4429  1 %sbttl 'locate_user - lookup user record in UAF'
;  4367     4430  1 routine locate_user (size, buffer, lock_record) =
;  4368     4431  2 begin
;  4369     4432  2
;  4370     4433  2 !++
;  4371     4434  2 !
;  4372     4435  2 ! FUNCTIONAL DESCRIPTION:
;  4373     4436  2 !
;  4374     4437  2 !       Routine to locate a user record in the UAF file.
;  4375     4438  2 !
;  4376     4439  2 ! INPUTS:
;  4377     4440  2 !
;  4378     4441  2 !       SIZE - size of the username string
;  4379     4442  2 !       BUFFER - address of the username string
;  4380     4443  2 !       LOCK_RECORD - specifies that the GET shall lock the record
;  4381     4444  2 !
;  4382     4445  2 ! IMPLICIT INPUTS:
;  4383     4446  2 !
;  4384     4447  2 !       UAFRAB - RMS data structure for SYSUAF.DAT
;  4385     4448  2 !
;  4386     4449  2 ! OUTPUTS:
;  4387     4450  2 !
;  4388     4451  2 !       none
;  4389     4452  2 !
;  4390     4453  2 ! IMPLICIT OUTPUTS:
;  4391     4454  2 !
;  4392     4455  2 !       If record is found, RECBUF contains the located record.
;  4393     4456  2 !
;  4394     4457  2 ! ROUTINE VALUE:
;  4395     4458  2 !
;  4396     4459  2 !       true -> record found
;  4397     4460  2 !       false -> record not found
;  4398     4461  2 !
;  4399     4462  2 ! SIDE EFFECTS:
;  4400     4463  2 !
;  4401     4464  2 !       none
;  4402     4465  2 !--
;  4403     4466  2
;  4404     4467  2 local
;  4405     4468  2         found;                                  ! record found indicator
;  4406     4469  2
;  4407     4470  2 found = true;                                   ! assume record was found
;  4408     4471  2
;  4409     4472  2 ch$copy (.size, .buffer, %char (' '),
;  4410     4473  2          uaf$s_username, recbuf[uaf$t_username]);
;  4411     4474  2
;  4412     4475  2 if .lock_record
;  4413     4476  2 then uafrab[rab$l_rop] = rab$m_rlk
;  4414     4477  2 else uafrab[rab$l_rop] = rab$m_rrl or rab$m_nlk;
;  4415     4478  2
;  4416     4479  3 if not (rmserr = get_uaf_record ())
;  4417     4480  2 then found = false;
;  4418     4481  2
;  4419     4482  2 return .found;                                  ! return indicator
;  4420     4483  1 end;
```

```
                                                00FC 00000 LOCATE_USER:
                                                           .WORD   Save R2,R3,R4,R5,R6,R7                          ; 4430
                                 57 00000000'   00  9E 00002        MOVAB   UAFRAB+4, R7                           ;
                                 56             01  D0 00009        MOVL    #1, FOUND                              ; 4470
            20          20    08 BC       04   AC  2C 0000C        MOVC5   SIZE, @BUFFER, #32, #32, RECBUF+4      ; 4473
                                       FA18   C7      00013                                                       ;
                                 09       0C   AC  E9 00016        BLBC    LOCK_RECORD, 1$                        ; 4475
                                 67 00080000   8F  D0 0001A        MOVL    #524288, UAFRAB+4                      ; 4476
                                               07  11 00021        BRB     2$                                     ;
                                 67 00100008   8F  D0 00023 1$:    MOVL    #1048584, UAFRAB+4                     ; 4477
                     00000000V   00            00  FB 0002A 2$:    CALLS   #0, GET_UAF_RECORD                     ; 4479
                         FA0C    C7            50  D0 00031        MOVL    R0, RMSERR                             ;
                                 02            50  E8 00036        BLBS    R0, 3$                                 ;
                                               56  D4 00039        CLRL    FOUND                                  ; 4480
                                 50            56  D0 0003B 3$:    MOVL    FOUND, R0                              ; 4482
                                               04  0003E          RET                                            ; 4483

; Routine Size:  63 bytes,    Routine Base:  $CODE$ + 26F1
```

```
;  4422          4484  1 %sbttl 'get_uaf_record - routine to deal with record locking'
;  4423          4485  1 routine get_uaf_record =
;  4424          4486  2 begin
;  4425          4487  2
;  4426          4488  2 !++
;  4427          4489  2 !
;  4428          4490  2 ! FUNCTIONAL DESCRIPTION:
;  4429          4491  2 !
;  4430          4492  2 !       A common routine to GET records from SYSUAF.DAT, deal with retries
;  4431          4493  2 !       when the record is locked.
;  4432          4494  2 !
;  4433          4495  2 ! INPUTS:
;  4434          4496  2 !
;  4435          4497  2 !       none
;  4436          4498  2 !
;  4437          4499  2 ! IMPLICIT INPUTS:
;  4438          4500  2 !
;  4439          4501  2 !       UAFRAB - RMS data structure for SYSUAF.DAT
;  4440          4502  2 !
;  4441          4503  2 ! OUTPUTS:
;  4442          4504  2 !
;  4443          4505  2 !       none
;  4444          4506  2 !
;  4445          4507  2 ! IMPLICIT OUTPUTS:
;  4446          4508  2 !
;  4447          4509  2 !       RECBUF - The user's record
;  4448          4510  2 !
;  4449          4511  2 ! ROUTINE VALUE:
;  4450          4512  2 !
;  4451          4513  2 !       RMS status code
;  4452          4514  2 !
;  4453          4515  2 ! SIDE EFFECTS:
;  4454          4516  2 !
;  4455          4517  2 !       none
;  4456          4518  2 !--
;  4457          4519  2
;  4458          4520  2 local
;  4459          4521  2         counter,                        ! number of retries remaining
;  4460          4522  2         success;
;  4461          4523  2
;  4462          4524  2 !
;  4463          4525  2 ! If anybody's record is locked it shouldn't remain that way for long.
;  4464          4526  2 ! Also, ignore the special system password record.
;  4465          4527  2 !
;  4466          4528  2
;  4467          4529  2 counter = retry_rlk;
;  4468          4530  4 while ( ((success = $get (rab = uafrab)) eql rms$_rlk)
;  4469          4531  3         or ch$eql(UAF$S_USERNAME, RECBUF[UAF$T_USERNAME],
;  4470          4532  3                 17, uplit(%ascii'<System+Password>'), %c' ') )
;  4471          4533  3       and ( (counter = .counter - 1) geq 0)
;  4472          4534  2 do
;  4473          4535  2     if $schdwk (daytim = wakedelta) then $hiber;
;  4474          4536  2
;  4475          4537  2 .success
;  4476          4538  1 end;
```

```
                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

72  6F  77  73  73  61  50  2B  6D  65  74  73  79  53  3C  00878 P.AFG:  .ASCII  \<System+Password>\<0><0><0>
                                        00  00  00  3E  64  00887

                                                                .PSECT  $CODE$,NOWRT,2

                                                        003C 00000 GET_UAF_RECORD:
                                                                .WORD   Save R2,R3,R4,R5                        ; 4485
                                    54                  08   D0 00002          MOVL    #8, COUNTER            ; 4529
                                                00000000'   00   9F 00005 1$:    PUSHAB  UAFRAB                 ; 4530
                                00000000G   00              01   FB 0000B          CALLS   #1, SYS$GET
                                    55                      50   D0 00012          MOVL    R0, SUCCESS
                                000182AA    8F              55   D1 00015          CMPL    SUCCESS, #98986
                                                            10   13 0001C          BEQL    2$
          11                20 00000000'  00              20   2D 0001E          CMPC5   #32, RECBUF+4, #32, #17, P.AFG  ; 4531
                                                00000000'   00        00027
                                                            21   12 0002C          BNEQ    3$
                                                            54   D7 0002E 2$:     DECL    COUNTER                ; 4533
                                                            1D   19 00030          BLSS    3$
                                                            7E   D4 00032          CLRL    -(SP)                  ; 4535
                                                00000000'   00   9F 00034          PUSHAB  WAKEDELTA
                                                            7E   7C 0003A          CLRQ    -(SP)
                                00000000G   00              04   FB 0003C          CALLS   #4, SYS$SCHDWK
                                    BF                      50   E9 00043          BLBC    R0, 1$
                                00000000G   00              00   FB 00046          CALLS   #0, SYS$HIBER
                                                            B6   11 0004D          BRB     1$
                                    50                      55   D0 0004F 3$:     MOVL    SUCCESS, R0            ; 4538
                                                                04 00052          RET

; Routine Size:  83 bytes,    Routine Base:  $CODE$ + 2730
```

```
;4478      4539  1 %sbttl 'get_cmd_line - input user command line'
;4479      4540  1 routine get_cmd_line =
;4480      4541  2 begin
;4481      4542  2
;4482      4543  2 !++
;4483      4544  2 !
;4484      4545  2 ! FUNCTIONAL DESCRIPTION:
;4485      4546  2 !
;4486      4547  2 !       This routine reads in the command line from the user,
;4487      4548  2 !       reading additional lines if continuation is specified by
;4488      4549  2 !       a '-' as the last character on the input line.
;4489      4550  2 !       A zero byte is inserted following the last input character read.
;4490      4551  2 !
;4491      4552  2 ! INPUTS:
;4492      4553  2 !
;4493      4554  2 !       none
;4494      4555  2 !
;4495      4556  2 ! IMPLICIT INPUTS:
;4496      4557  2 !
;4497      4558  2 !       CMDBUF - buffer to receive the user's command line
;4498      4559  2 !       CMDBUFLEN - literal length of CMDBUF
;4499      4560  2 !
;4500      4561  2 ! OUTPUTS:
;4501      4562  2 !
;4502      4563  2 !       none
;4503      4564  2 !
;4504      4565  2 ! IMPLICIT OUTPUTS:
;4505      4566  2 !
;4506      4567  2 !       CMDBUF is filled with command line
;4507      4568  2 !
;4508      4569  2 ! ROUTINE VALUE:
;4509      4570  2 !
;4510      4571  2 !       none
;4511      4572  2 !
;4512      4573  2 ! SIDE EFFECTS:
;4513      4574  2 !
;4514      4575  2 !       none
;4515      4576  2 !--
;4516      4577  2
;4517      4578  2 map
;4518      4579  2         cmdlindsc: vector;
;4519      4580  2
;4520      4581  2 local
;4521      4582  2         ptr,                                  ! index into CMDBUF
;4522      4583  2         buflen;                               ! remaining buffer length
;4523      4584  2
;4524      4585  2 !
;4525      4586  2 ! Prompt with normal prompt string until some input is supplied.
;4526      4587  2 !
;4527      4588  2
;4528      4589  2 do
;4529      4590  2     ask (accprmpt, cmdbuf[0], cmdbuflen)
;4530      4591  2 until
;4531      4592  2     .insize neq 0;
;4532      4593  2
;4533      4594  2 !
;4534      4595  2 ! Now that line has been read, continue reading until
```

```
;4535          4596  2 ! last character is not a '-'.
;4536          4597  2 !
;4537          4598  2
;4538          4599  2 ptr = .insize - 1;                          ! index to last character read
;4539          4600  2 buflen = cmdbuflen;                        ! initial buffer size
;4540          4601  2
;4541          4602  2 while
;4542          4603  2     .cmdbuf[.ptr] eql '-'
;4543          4604  2 do
;4544          4605  3     begin
;4545          4606  3
;4546          4607  3     !
;4547          4608  3     ! Read starting at the position of the '-'. Then adjust the
;4548          4609  3     ! index and remaining length to point to the last character
;4549          4610  3     ! of the next input.
;4550          4611  3     !
;4551          4612  3
;4552          4613  3     buflen = .buflen - (.insize + 1);
;4553          4614  3     ask (accprmpt2, cmdbuf[.ptr], .buflen);
;4554          4615  3     if (.ptr + .insize)  lss  cmdbufmax
;4555          4616  3     then
;4556          4617  3         ptr = .ptr + .insize - 1
;4557          4618  3     else
;4558          4619  4         begin
;4559          4620  4         LIB$SIGNAL(UAF$_CMDTOOLONG, 1, cmdbufmax);
;4560          4621  4         return false;
;4561          4622  3         end;
;4562          4623  2     end;
;4563          4624  2
;4564          4625  2
;4565          4626  2 cmdlindsc [0] = .ptr + 1;
;4566          4627  2 cmdlindsc [1] = cmdbuf;
;4567          4628  2
;4568          4629  2 return true;
;4569          4630  2
;4570          4631  1 end;


                                    003C 00000 GET_CMD_LINE:
                                               .WORD   Save R2,R3,R4,R5                ;4540
                55 00000000V    00  9E 00002           MOVAB   ASK, R5
                54 00000000'    00  9E 00009           MOVAB   CMDBUF, R4
                7E      0400    8F  3C 00010 1$:       MOVZWL  #1024, -(SP)            ;4590
                                54  DD 00015           PUSHL   R4
                   00000000'    00  9F 00017           PUSHAB  ACCPRMPT
                65              03  FB 0001D           CALLS   #3, ASK
                50      0990    C4  D0 00020           MOVL    INSIZE, R0              ;4592
                                E9  13 00025           BEQL    1$
                52        FF    A0  9E 00027           MOVAB   -1(R0), PTR             ;4599
                53      0400    8F  3C 0002B           MOVZWL  #1024, BUFLEN           ;4600
          51    52              54  C1 00030 2$:       ADDL3   R4, PTR, R1             ;4603
                2D              61  91 00034           CMPB    (R1), #45
                                40  12 00037           BNEQ    4$
          50    53      0990    C4  C3 00039           SUBL3   INSIZE, BUFLEN, R0      ;4613
```

```
                                53       FF    A0  9E 0003F          MOVAB   -1(R0), BUFLEN                                    ;
                                                 0A  BB 00043          PUSHR   #^M<R',R3>                                      ; 4614
                                   00000000'     00  9F 00045          PUSHAB  ACCPRMPT2
                                65               03  FB 0004B          CALLS   #3, ASK
                          50    52       0990    C4  C1 0004E          ADDL3   INSIZE, PTR, R0                                 ; 4615
                     000001FC   8F               50  D1 00054          CMPL    R0, #508
                                                 06  18 0005B          BGEQ    3$
                                52       FF    A0  9E 0005D          MOVAB   -1(R0), PTR                                     ; 4617
                                                 CD  11 00061          BRB     2$
                                7E       01FC  8F  3C 00063 3$:      MOVZWL  #508, -(SP)                                     ; 4620
                                                 01  DD 00068          PUSHL   #1
                                   00000000G     8F  DD 0006A          PUSHL   #UAF$_CMDTOOLONG
                     00000000G  00               03  FB 00070          CALLS   #3, LIB$SIGNAL
                                                 13  11 00077          BRB     5$                                              ; 4621
                     00000000'  00       01      A2  9E 00079 4$:      MOVAB   1(R2), CMDLINDSC                                ; 4626
                     00000000'  00               64  9E 00081          MOVAB   CMDBUF, CMDLINDSC+4                             ; 4627
                                50               01  D0 00088          MOVL    #1, R0                                          ; 4629
                                                 04  0008B            RET
                                                 50  D4 0008C 5$:      CLRL    R0                                              ; 4631
                                                 04  0008E            RET                                                     ;

; Routine Size:  143 bytes,    Routine Base:  $CODE$ + 2783
```

```
;  4572          4632  1 %sbttl 'ask - prompt terminal for input'
;  4573          4633  1 global routine ask (string, buffer, len) : novalue =
;  4574          4634  2 begin
;  4575          4635  2
;  4576          4636  2 !++
;  4577          4637  2 !
;  4578          4638  2 ! FUNCTIONAL DESCTIPTION:
;  4579          4639  2 !
;  4580          4640  2 !       Routine to prompt user for input.  The input string is read
;  4581          4641  2 !       into the user specified buffer and the size read
;  4582          4642  2 !       is placed in the global INSIZE.
;  4583          4643  2 !
;  4584          4644  2 ! INPUTS:
;  4585          4645  2 !
;  4586          4646  2 !       STRING - the address of a counted ascii prompt string
;  4587          4647  2 !       BUFFER - address of the input buffer
;  4588          4648  2 !       LEN - length of the input buffer
;  4589          4649  2 !
;  4590          4650  2 ! IMPLICIT INPUTS:
;  4591          4651  2 !
;  4592          4652  2 !       none
;  4593          4653  2 !
;  4594          4654  2 ! OUTPUTS:
;  4595          4655  2 !
;  4596          4656  2 !       none
;  4597          4657  2 !
;  4598          4658  2 ! IMPLICIT OUTPUTS:
;  4599          4659  2 !
;  4600          4660  2 !       INSIZE - size of the input string
;  4601          4661  2 !
;  4602          4662  2 ! ROUTINE VALUE:
;  4603          4663  2 !
;  4604          4664  2 !       none
;  4605          4665  2 !
;  4606          4666  2 ! SIDE EFFECTS:
;  4607          4667  2 !
;  4608          4668  2 !       none
;  4609          4669  2 !--
;  4610          4670  2
;  4611          4671  2 map
;  4612          4672  2         buffer  : ref vector[,byte];
;  4613          4673  2
;  4614          4674  2 inrab[rab$l_pbf] = .string + 1;                    ! prompt string address
;  4615          4675  2 inrab[rab$b_psz] = .(.string)<0,8>;                ! prompt size
;  4616          4676  2 inrab[rab$l_ubf] = .buffer;                        ! buffer address
;  4617          4677  2 inrab[rab$w_usz] = .len;                           ! buffer size
;  4618          4678  2
;  4619          4679  2 !
;  4620          4680  2 ! If end of file encountered on get (either ^Z from terminal or
;  4621          4681  2 ! end of file on indirect command file) then take exit path.
;  4622          4682  2 !
;  4623          4683  2
;  4624          4684  2 if $get (rab=inrab) eql rms$_eof
;  4625          4685  2 then exit_uaf ();
;  4626          4686  2
;  4627          4687  2 if (insize = .inrab[rab$w_rsz]) neq 0                ! get input size
;  4628          4688  2 then
```

```
;                ask - prompt terminal for input

; 4629          4689  3     begin
; 4630          4690  3     incru i to .insize - 1                          ! Upcasing is done here because the CVT
; 4631          4691  3     do                                              ! option does not work under batch.
; 4632          4692  4         begin
; 4633          4693  4         if .buffer[.i] gequ 'a' and .buffer[.i] lequ 'z'
; 4634          4694  4         then buffer[.i] = .buffer[.i] and not %o'040';
; 4635          4695  3         end;
; 4636          4696  2     end;
; 4637          4697  1 end;


                                                0004 00000         .ENTRY  ASK, Save R2                                    ; 4633
                                52 00000000'    00   9E 00002      MOVAB   INSIZE, R2                                      ;
                0090    C2        04    AC      01   C1 00009      ADDL3   #1, STRING, INRAB+48                            ; 4674
                                0094    C2  04    BC   90 00010      MOVB    @STRING, INRAB+52                           ; 4675
                                0084    C2  08    AC   D0 00016      MOVL    BUFFER, INRAB+36                            ; 4676
                                0080    C2  0C    AC   B0 0001C      MOVW    LEN, INRAB+32                               ; 4677
                                            60    A2   9F 00022      PUSHAB  INRAB                                       ; 4684
                            00000000G   00        01   FB 00025      CALLS   #1, SYS$GET                                 ;
                            0001827A    8F        50   D1 0002C      CMPL    R0, #98938                                  ;
                                                  07   12 00033      BNEQ    1$                                          ;
                            00000000V   00        00   FB 00035      CALLS   #0, EXIT_UAF                                ; 4685
                                        62  0082  C2   3C 0003C 1$:   MOVZWL  INRAB+34, INSIZE                            ; 4687
                                                  24   13 00041      BEQL    5$                                          ;
                        51              62        01   C3 00043      SUBL3   #1, INSIZE, R1                              ; 4690
                                                  50   D4 00047      CLRL    I                                           ;
                                                  17   11 00049      BRB     4$                                          ;
                                61      8F    08 BC40  91 0004B 2$:   CMPB    @BUFFER[I], #97                             ; 4693
                                                  0D   1F 00051      BLSSU   3$                                          ;
                                7A      8F    08 BC40  91 00053      CMPB    @BUFFER[I], #122                            ;
                                                  05   1A 00059      BGTRU   3$                                          ;
                            08 BC40             20   8A 0005B      BICB2   #32, @BUFFER[I]                             ; 4694
                                                  50   D6 00060 3$:   INCL    I                                           ; 4690
                                    51            50   D1 00062 4$:   CMPL    I, R1                                       ;
                                                  E4   1B 00065      BLEQU   2$                                          ;
                                                       04 00067 5$:   RET                                                 ; 4697

; Routine Size:  104 bytes,    Routine Base:  $CODE$ + 2812
```

```
;4639        4698  1 %sbttl 'fmt_sys_msg - output system message file message'
;4640        4699  1 global routine fmt_sys_msg (faostr, msgid, p1) : novalue =
;4641        4700  2 begin
;4642        4701  2
;4643        4702  2 !++
;4644        4703  2 !
;4645        4704  2 ! FUNCTIONAL DESCRIPTION:
;4646        4705  2 !
;4647        4706  2 !       This routine outputs an error message followed by
;4648        4707  2 !       the text found in the system message file for the
;4649        4708  2 !       error condition.  If the message is not found, the message
;4650        4709  2 !       ID itself is printed.
;4651        4710  2 !
;4652        4711  2 ! INPUTS:
;4653        4712  2 !
;4654        4713  2 !       FAOSTR - address of counted ascii message to be printed
;4655        4714  2 !       MSGID  - error number
;4656        4715  2 !       P1     - the first of possibly several parameters to FAO
;4657        4716  2 !
;4658        4717  2 ! IMPLICIT INPUTS:
;4659        4718  2 !
;4660        4719  2 !       None
;4661        4720  2 !
;4662        4721  2 ! OUTPUTS:
;4663        4722  2 !
;4664        4723  2 !       None
;4665        4724  2 !
;4666        4725  2 ! IMPLICIT OUTPUTS:
;4667        4726  2 !
;4668        4727  2 !       None
;4669        4728  2 !
;4670        4729  2 ! ROUTINE VALUE:
;4671        4730  2 !
;4672        4731  2 !       None
;4673        4732  2 !
;4674        4733  2 ! SIDE EFFECTS:
;4675        4734  2 !
;4676        4735  2 !       None
;4677        4736  2 !--
;4678        4737  2
;4679        4738  2 local
;4680        4739  2         buffer  : vector [200, byte],           ! buffer to receive message
;4681        4740  2         bufdsc  : vector [2, long],             ! string descriptor
;4682        4741  2         code;                                   ! save return code
;4683        4742  2
;4684        4743  2 bufdsc[0] = 200;                                ! construct string descriptor
;4685        4744  2 bufdsc[1] = buffer;
;4686        4745  2
;4687        4746  2 code = $getmsg (msgid = .msgid, msglen = bufdsc[0], bufadr = bufdsc[0]);
;4688        4747  2
;4689        4748  2 !
;4690        4749  2 ! Output internal message.  Then output system error or error number.
;4691        4750  2 !
;4692        4751  2 faodsc[dsc$w_length] = . (.faostr)<0,8>;        ! input string descriptor
;4693        4752  2 faodsc[dsc$a_pointer] = .faostr+1;
;4694      P 4753  2 $faol (ctrstr = faodsc,
;4695      P 4754  2         outlen = outrab[rab$w_rsz],
```

```
;  4696        P 4755  2          outbuf = disdsc,
;  4697          4756  2          prmlst = p1);
;  4698          4757  2
;  4699          4758  2 $put (rab = outrab);
;  4700          4759  2
;  4701          4760  2 if .code eql ss$_msgnotfnd
;  4702          4761  2 then LIB$SIGNAL(UAF$_SYSMSG2, 1, .msgid)
;  4703          4762  2 else LIB$SIGNAL(UAF$_SYSMSG1, 1, bufdsc[0]);
;  4704          4763  1 end;


                                                                .EXTRN  SYS$GETMSG, SYS$FAOL

                                                 000C 00000     .ENTRY  FMT_SYS_MSG, Save R2,R3                         ;  4699
                                  53 00000000'   00   9E 00002  MOVAB   FAODSC, R3                                      ;
                                  5E     FF34    CE   9E 00009  MOVAB   -204(SP), SP                                   ;
                                  7E       C8    8F   9A 0000E  MOVZBL  #200, BUFDSC                                   ;  4743
                            04    AE       08    AE   9E 00012  MOVAB   BUFFER, BUFDSC+4                               ;  4744
                                  7E             0F   7D 00017  MOVQ    #15, -(SP)                                    ;  4746
                                           08    AE   9F 0001A  PUSHAB  BUFDSC                                        ;
                                           0C    AE   9F 0001D  PUSHAB  BUFDSC                                        ;
                                           08    AC   DD 00020  PUSHL   MSGID                                         ;
                       00000000G   00              05   FB 00023  CALLS   #5, SYS$GETMSG                              ;
                                  52               50   D0 0002A  MOVL    R0, CODE                                    ;
                                  63         04    BC   9B 0002D  MOVZBW  @FAOSTR, FAODSC                             ;  4751
                      04    A3        04    AC        01   C1 00031  ADDL3   #1, FAOSTR, FAODSC+4                     ;  4752
                                           0C    AC   9F 00037  PUSHAB  P1                                           ;  4756
                                           F8    A3   9F 0003A  PUSHAB  DISDSC                                       ;
                                        072E     C3   9F 0003D  PUSHAB  OUTRAB+34                                    ;
                                                 53   DD 00041  PUSHL   R3                                          ;
                       00000000G   00              04   FB 00043  CALLS   #4, SYS$FAOL                               ;
                                        070C     C3   9F 0004A  PUSHAB  OUTRAB                                       ;  4758
                       00000000G   00            01   FB 0004E  CALLS   #1, SYS$PUT                                ;
                       00000621    8F            52   D1 00055  CMPL    CODE, #1569                                 ;  4760
                                                 0D   12 0005C  BNEQ    1$                                          ;
                                           08    AC   DD 0005E  PUSHL   MSGID                                        ;  4761
                                                 01   DD 00061  PUSHL   #1                                          ;
                                  00000000G      8F   DD 00063  PUSHL   #UAF$_SYSMSG2                               ;
                                                 0A   11 00069  BRB     2$                                          ;
                                                 5E   DD 0006B  1$:     PUSHL   SP                                  ;  4762
                                                 01   DD 0006D  PUSHL   #1                                          ;
                                  00000000G      8F   DD 0006F  PUSHL   #UAF$_SYSMSG1                               ;
                       00000000G   00            03   FB 00075  2$:     CALLS   #3, LIB$SIGNAL                      ;
                                                      04 0007C  RET                                                  ;  4763

; Routine Size:  125 bytes,    Routine Base:  $CODE$ + 287A
```

```
;  4706          4764  1 %sbttl 'faoout - output formatted message'
;  4707          4765  1 global routine faoout (string, p1) =
;  4708          4766  2 begin
;  4709          4767  2
;  4710          4768  2 !++
;  4711          4769  2 !
;  4712          4770  2 ! FUNCTIONAL DESCRIPTION:
;  4713          4771  2 !
;  4714          4772  2 !       Routine to output a formatted string.
;  4715          4773  2 !
;  4716          4774  2 ! INPUTS:
;  4717          4775  2 !
;  4718          4776  2 !       STRING - address of a counted ASCII FAO control string.
;  4719          4777  2 !       P1 - the first of possibly several parameters to FAO
;  4720          4778  2 !
;  4721          4779  2 ! IMPLICIT INPUTS:
;  4722          4780  2 !
;  4723          4781  2 !       none
;  4724          4782  2 !
;  4725          4783  2 ! OUTPUTS:
;  4726          4784  2 !
;  4727          4785  2 !       none
;  4728          4786  2 !
;  4729          4787  2 ! IMPLICIT OUTPUTS:
;  4730          4788  2 !
;  4731          4789  2 !       none
;  4732          4790  2 !
;  4733          4791  2 ! ROUTINE VALUE:
;  4734          4792  2 !
;  4735          4793  2 !       FAOOUT always returns FALSE, as it is often used on the
;  4736          4794  2 !       return from an error condition.
;  4737          4795  2 !
;  4738          4796  2 ! SIDE EFFECTS:
;  4739          4797  2 !
;  4740          4798  2 !       none
;  4741          4799  2 !--
;  4742          4800  2
;  4743          4801  2
;  4744          4802  2 faodsc[dsc$w_length] = . (.string)<0,8>;        ! input string descriptor
;  4745          4803  2 faodsc[dsc$a_pointer] = .string+1;
;  4746        P 4804  2 $faol (ctrstr = faodsc,
;  4747        P 4805  2        outlen = outrab[rab$w_rsz],
;  4748        P 4806  2        outbuf = disdsc,
;  4749          4807  2        prmlst = p1);
;  4750          4808  2
;  4751          4809  2 $put (rab = outrab);
;  4752          4810  2
;  4753          4811  2 false
;  4754          4812  1 end;


                                        0004 00000          .ENTRY  FAOOUT, Save R2                         ; 4765
                            52 00000000'  00   9E 00002     MOVAB   FAODSC, R2
                            62        04   BC   9B 00009     MOVZBW  @STRING, FAODSC                         ; 4802
```

```
                04   A2        04   AC                01   C1 0000D          ADDL3   #1, STRING, FAODSC+4                    ; 4803
                                                  08   AC   9F 00013          PUSHAB  P1                                      ; 4807
                                                  F8   A2   9F 00016          PUSHAB  DISDSC                                  ;
                                               072E    C2   9F 00019          PUSHAB  OUTRAB+34                               ;
                                                      52   DD 0001D          PUSHL   R2                                      ;
                         00000000G  00                04   FB 0001F          CALLS   #4, SYS$FAOL                            ;
                                                  070C   C2   9F 00026          PUSHAB  OUTRAB                                  ; 4809
                         00000000G  00                01   FB 0002A          CALLS   #1, SYS$PUT                             ;
                                                      50   D4 00031          CLRL    R0                                      ; 4812
                                                           04 00033          RET                                             ;

; Routine Size:  52 bytes,    Routine Base:  $CODE$ + 28F7
```

```
;  4756          4813  1 %sbttl 'help_uaf - help routine'
;  4757          4814  1 global routine help_uaf : novalue =
;  4758          4815  2 begin
;  4759          4816  2
;  4760          4817  2 !++
;  4761          4818  2 !
;  4762          4819  2 ! FUNCTIONAL DESCRIPTION:
;  4763          4820  2 !
;  4764          4821  2 !       Print out the help message or messages.
;  4765          4822  2 !
;  4766          4823  2 ! INPUTS:
;  4767          4824  2 !
;  4768          4825  2 !       none
;  4769          4826  2 !
;  4770          4827  2 ! OUTPUTS:
;  4771          4828  2 !
;  4772          4829  2 !       none
;  4773          4830  2 !
;  4774          4831  2 ! ROUTINE VALUE:
;  4775          4832  2 !
;  4776          4833  2 !       none
;  4777          4834  2 !
;  4778          4835  2 ! SIDE EFFECTS:
;  4779          4836  2 !
;  4780          4837  2 !       none
;  4781          4838  2 !--
;  4782          4839  2
;  4783          4840  2 map
;  4784          4841  2         cmdlindsc: vector;
;  4785          4842  2
;  4786          4843  2 local
;  4787          4844  2         line_dsc        : vector [2];
;  4788          4845  2
;  4789          4846  2 line_dsc[0] = .cmdlindsc[0];
;  4790          4847  2 line_dsc[1] = .cmdlindsc[1];
;  4791          4848  2
;  4792          4849  2 !
;  4793          4850  2 ! The first thing to do is to remove 'help' from the command line
;  4794          4851  2 ! and give the help routine the remainder of the command line.
;  4795          4852  2 ! Find the beginning of the word 'help'.
;  4796          4853  2 !
;  4797          4854  2 if .(.line_dsc[1])<0,8> eql ' '
;  4798          4855  2 then
;  4799          4856  3     begin
;  4800          4857  3     line_dsc[1] = ch$find_not_ch (.line_dsc[0], .line_dsc[1], ' ');
;  4801          4858  3     line_dsc[0] = .line_dsc[0] - (.line_dsc[1] - .cmdlindsc[1]);
;  4802          4859  2     end;
;  4803          4860  2
;  4804          4861  2 !
;  4805          4862  2 ! Now skip past the 'help' to the first blank (if there is one)
;  4806          4863  2 !
;  4807          4864  2 if ch$fail (line_dsc[1] = ch$find_ch (.line_dsc[0], .line_dsc[1], ' '))
;  4808          4865  2 then                                                                    ;
;  4809          4866  2     !
;  4810          4867  2     ! No blank, set empty string
;  4811          4868  2     !
;  4812          4869  2     line_dsc[0] = 0
```

```
;  4813          4870  2 else
;  4814          4871  2     !
;  4815          4872  2     ! Found a blank, set pointer to it
;  4816          4873  2     !
;  4817          4874  2     line_dsc[0] = .cmdlindsc[0] - (.line_dsc[1] - .cmdlindsc[1]);
;  4818          4875  2
;  4819          4876  2 !
;  4820          4877  2 ! Start an interactive HELP session
;  4821          4878  2 !
;  4822          4879  2
;  4823          4880  2 if not lbr$output_help (lib$put_output, 0, line_dsc,
;  4824          4881  2                          $descriptor ('uafhelp'), 0,
;  4825          4882  2                          lib$get_input)
;  4826          4883  2 then LIB$SIGNAL(UAF$_HELPERR);
;  4827          4884  2
;  4828          4885  1 end;


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                70  6C  65  68  66  61  75  0088C P.AFI:  .ASCII  \uafhelp\                                ;
                                                        00893           .BLKB   1
                                            00000007    00894 P.AFH:    .LONG   7                                        ;
                                            00000000'   00898           .ADDRESS P.AFI                                   ;


                                                                .PSECT  $CODE$,NOWRT,2

                                                    000C 00000          .ENTRY  HELP_UAF, Save R2,R3                     ; 4814
                                        5E                  04  C2  00002    SUBL2   #4, SP                               ;
                                        53  00000000'       00  D0  00005    MOVL    CMDLINDSC, R3                        ; 4846
                                                            53  DD  0000C    PUSHL   R3                                   ;
                                        52  00000000'       00  D0  0000E    MOVL    CMDLINDSC+4, R2                      ; 4847
                                    04  AE                  52  D0  00015    MOVL    R2, LINE_DSC+4                       ;
                                    20          04      BE  91  00019        CMPB    @LINE_DSC+4, #32                     ; 4854
                                                        15  12  0001D        BNEQ    2$
                            04  BE                  6E  20  3B  0001F        SKPC    #32, LINE_DSC, @LINE_DSC+4           ; 4857
                                                        02  12  00024        BNEQ    1$
                                                        51  D4  00026        CLRL    R1
                                    04  AE                  51  D0  00028 1$:  MOVL    R1, LINE_DSC+4
                                50              52          04  AE  C3  0002C    SUBL3   LINE_DSC+4, R2, R0               ; 4858
                                                6E          50  C0  00031    ADDL2   R0, LINE_DSC
                            04  BE                  6E      20  3A  00034 2$:  LOCC    #32, LINE_DSC, @LINE_DSC+4           ; 4864
                                                        02  12  00039        BNEQ    3$
                                                        51  D4  0003B        CLRL    R1
                                    04  AE                  51  D0  0003D 3$:  MOVL    R1, LINE_DSC+4
                                                        04  12  00041        BNEQ    4$
                                                        6E  D4  00043        CLRL    LINE_DSC                             ; 4869
                                                        09  11  00045        BRB     5$
                                50              52          04  AE  C3  00047 4$:  SUBL3   LINE_DSC+4, R2, R0           ; 4874
                                6E              50          53  C1  0004C    ADDL3   R3, R0, LINE_DSC
                                            000000006       00  9F  00050 5$:  PUSHAB  LIB$GET_INPUT                     ; 4880
                                                            7E  D4  00056    CLRL    -(SP)
                                            00000000'       00  9F  00058    PUSHAB  P.AFH                                ; 4881
                                                    0C      AE  9F  0005E    PUSHAB  LINE_DSC                             ; 4880
```

```
                                 7E  D4 00061          CLRL    -(SP)                                           ;
                    00000000G    00  9F 00063          PUSHAB  LIB$PUT_OUTPUT                                  ;
                 00000000G  00       06  FB 00069      CALLS   #6, LBR$OUTPUT_HELP                             ;
                            0D       50  E8 00070      BLBS    R0, 6$                                          ;
                    00000000G    8F  DD 00073          PUSHL   #UAF$_HELPERR                                   ; 4883
                 00000000G  00       01  FB 00079      CALLS   #1, LIB$SIGNAL                                  ;
                                     04 00080 6$:       RET                                                     ; 4885

; Routine Size:  129 bytes,    Routine Base:  $CODE$ + 292B
```

```
;  4830          4886  1 %sbttl 'exit_uaf - normal exit routine'
;  4831          4887  1 global routine exit_uaf : novalue =
;  4832          4888  2 begin
;  4833          4889  2
;  4834          4890  2 !++
;  4835          4891  2 !
;  4836          4892  2 ! FUNCTIONAL DESCRIPTION:
;  4837          4893  2 !
;  4838          4894  2 !       Normal exit routine.
;  4839          4895  2 !
;  4840          4896  2 ! INPUTS:
;  4841          4897  2 !
;  4842          4898  2 !       none
;  4843          4899  2 !
;  4844          4900  2 ! OUTPUTS:
;  4845          4901  2 !
;  4846          4902  2 !       none
;  4847          4903  2 !
;  4848          4904  2 ! ROUTINE VALUE:
;  4849          4905  2 !
;  4850          4906  2 !       none
;  4851          4907  2 !
;  4852          4908  2 ! SIDE EFFECTS:
;  4853          4909  2 !
;  4854          4910  2 !       image exits
;  4855          4911  2 !--
;  4856          4912  2
;  4857          4913  2 $close (fab = uaffab);
;  4858          4914  2
;  4859          4915  2 !
;  4860          4916  2 ! Inform user of file modifications.
;  4861          4917  2 !
;  4862          4918  2
;  4863          4919  2 if .modify_flag
;  4864          4920  2 then
;  4865          4921  2 !
;  4866          4922  2 ! File has been modified
;  4867          4923  2 !
;  4868          4924  2     LIB$SIGNAL(UAF$_DONEMSG)            ! tell user all is done
;  4869          4925  2 else
;  4870          4926  2 !
;  4871          4927  2 ! Here, no modifications were made to file.
;  4872          4928  2 !
;  4873          4929  2     LIB$SIGNAL(UAF$_NOMODS);
;  4874          4930  2
;  4875          4931  2 if .netuaf_exists
;  4876          4932  2 then
;  4877          4933  3     begin
;  4878          4934  3     if not .netuaf_modified
;  4879          4935  3     then
;  4880          4936  3         LIB$SIGNAL(UAF$_NAFNOMODS)
;  4881          4937  3     else
;  4882          4938  3         LIB$SIGNAL(UAF$_NAFDONEMSG);
;  4883          4939  2     end;
;  4884          4940  2
;  4885          4941  2 if .rdb_exists
;  4886          4942  2 then
```

```
;  4887          4943  3     begin
;  4888          4944  3     if .rightslist_modified
;  4889          4945  3     then
;  4890          4946  3     !
;  4891          4947  3     ! File has been modified
;  4892          4948  3     !
;  4893          4949  3         LIB$SIGNAL(UAF$_RDBDONEMSG)             ! tell user all is done
;  4894          4950  3     else
;  4895          4951  3     !
;  4896          4952  3     ! Here, no modifications were made to file.
;  4897          4953  3     !
;  4898          4954  3         LIB$SIGNAL(UAF$_RDBNOMODS);
;  4899          4955  2     end ;
;  4900          4956  2
;  4901          4957  2 $exit (code = true);
;  4902          4958  1 end;


                                                        .EXTRN  SYS$EXIT

                                        001C 00000      .ENTRY  EXIT_UAF, Save R2,R3,R4                 ; 4887
                    54 00000000'    00   9E 00002      MOVAB   UAFFAB, R4                              ;
                    53 00000000'    00   9E 00009      MOVAB   NETUAF_EXISTS, R3                       ;
                    52 00000000G    00   9E 00010      MOVAB   LIB$SIGNAL, R2                          ;
                                    54   DD 00017      PUSHL   R4                                      ; 4913
              00000000G   00              01   FB 00019      CALLS   #1, SYS$CLOSE                           ;
                    08       F154         C4   E9 00020      BLBC    MODIFY_FLAG, 1$                 ; 4919
                       00000000G    8F     DD 00025      PUSHL   #UAF$_DONEMSG                   ; 4924
                                    06   11 0002B      BRB     2$                              ;
                       00000000G    8F   DD 0002D 1$:    PUSHL   #UAF$_NOMODS                    ; 4929
                 62                 01   FB 00033 2$:    CALLS   #1, LIB$SIGNAL                  ;
                 16                 63   E9 00036        BLBC    NETUAF_EXISTS, 5$               ; 4931
                 08       F158      C4   E8 00039        BLBS    NETUAF_MODIFIED, 3$             ; 4934
                       00000000G    8F   DD 0003E        PUSHL   #UAF$_NAFNOMODS                 ; 4936
                                    06   11 00044        BRB     4$                              ;
                       00000000G    8F   DD 00046 3$:    PUSHL   #UAF$_NAFDONEMSG                ; 4938
                 62                 01   FB 0004C 4$:    CALLS   #1, LIB$SIGNAL                  ;
                 15           04    A3   E9 0004F 5$:    BLBC    RDB_EXISTS, 8$                  ; 4941
                 08           0C    A3   E9 00053        BLBC    RIGHTSLIST_MODIFIED, 6$         ; 4944
                       00000000G    8F   DD 00057        PUSHL   #UAF$_RDBDONEMSG                ; 4949
                                    06   11 0005D        BRB     7$                              ;
                       00000000G    8F   DD 0005F 6$:    PUSHL   #UAF$_RDBNOMODS                 ; 4954
                 62                 01   FB 00065 7$:    CALLS   #1, LIB$SIGNAL                  ;
                                    01   DD 00068 8$:    PUSHL   #1                              ; 4957
              00000000G   00              01   FB 0006A        CALLS   #1, SYS$EXIT                    ;
                                        04 00071        RET                                     ; 4958

; Routine Size:  114 bytes,    Routine Base:  $CODE$ + 29AC
```

```
;  4904          4959  1 %sbttl 'SIGNAL_SYNTAX - Report missing qualifier'
;  4905          4960  1 global routine SIGNAL_SYNTAX: novalue =
;  4906          4961  2 begin
;  4907          4962  2
;  4908          4963  2     LIB$SIGNAL(UAF$_ZISQUAL);
;  4909          4964  2
;  4910          4965  1 end;


                                        0000 00000          .ENTRY  SIGNAL_SYNTAX, Save nothing                     ; 4960
                              00000000G 8F   DD 00002       PUSHL   #UAF$_ZISQUAL                                   ; 4963
                    00000000G   00          01   FB 00008  CALLS   #1, LIB$SIGNAL                                  ;
                                            04 0000F        RET                                                     ; 4965

; Routine Size:  16 bytes,    Routine Base:  $CODE$ + 2A1E
```

```
;  4912          4966  1 %sbttl 'acc$exit - exit and cleanup routine'
;  4913          4967  1 routine acc$exit : novalue =
;  4914          4968  2 begin
;  4915          4969  2
;  4916          4970  2 !++
;  4917          4971  2 !
;  4918          4972  2 ! FUNCTIONAL DESCRIPTION:
;  4919          4973  2 !
;  4920          4974  2 !       Exit on error condition.
;  4921          4975  2 !
;  4922          4976  2 ! INPUTS:
;  4923          4977  2 !
;  4924          4978  2 !       none
;  4925          4979  2 !
;  4926          4980  2 ! OUTPUTS:
;  4927          4981  2 !
;  4928          4982  2 !       none
;  4929          4983  2 !--
;  4930          4984  2
;  4931          4985  2 $exit ();
;  4932          4986  1 end;


                                0000 00000 ACC$EXIT:
                                                .WORD   Save nothing                    ; 4967
                                01  DD 00002           PUSHL   #1                      ; 4985
                    00000000G   00  01  FB 00004        CALLS   #1, SYS$EXIT            ;
                                    04 0000B            RET                             ; 4986

; Routine Size:  12 bytes,    Routine Base:  $CODE$ + 2A2E
```

```
;  4934          4987  1 %sbttl 'UAF$MOD_SYS_PWD -- modify the system password'
;  4935          4988  1 global routine UAF$MOD_SYS_PWD: novalue =
;  4936          4989  2 begin
;  4937          4990  2
;  4938          4991  2 !++
;  4939          4992  2 !
;  4940          4993  2 ! Functional Description:
;  4941          4994  2 !
;  4942          4995  2 !       Modify the system password record by doing a $PUT on the UAF
;  4943          4996  2 !       with the UIF bit set for the username "<System+Password>"
;  4944          4997  2 !
;  4945          4998  2 !--
;  4946          4999  2
;  4947          5000  2 local
;  4948          5001  2
;  4949          5002  2     RBF,
;  4950          5003  2     PWD: block[DSC$K_D_BLN, byte]
;  4951          5004  2          preset([DSC$B_CLASS] = DSC$K_CLASS_D),
;  4952          5005  2     ROP;
;  4953          5006  2
;  4954          5007  2 !
;  4955          5008  2 !       Save current settings
;  4956          5009  2 !
;  4957          5010  2
;  4958          5011  2 RBF = .UAFRAB[RAB$L_RBF];
;  4959          5012  2 ROP = .UAFRAB[RAB$L_ROP];
;  4960          5013  2
;  4961          5014  2 !
;  4962          5015  2 !       Get new password and encrypt it
;  4963          5016  2 !
;  4964          5017  2
;  4965          5018  2 CLI$GET_VALUE(%ascid'SYSTEM_PASSWORD', PWD);
;  4966          5019  2
;  4967          5020  2 if .PWD[DSC$W_LENGTH] neq 0 then
;  4968          5021  2     LGI$HPWD(REC_ENCRYPT_DSC, PWD, UAF$C_PURDY_V, 0, %ASCID'<System+Password>')
;  4969          5022  2 else
;  4970          5023  2     ch$fill(0, UAF$S_PWD, RECBUF[UAF$Q_PWD]);
;  4971          5024  2
;  4972          5025  2 !
;  4973          5026  2 !       Fill in fields of the UAF record
;  4974          5027  2 !
;  4975          5028  2
;  4976          5029  2 ch$copy(17, UPLIT('<System+Password>'), ' ',
;  4977          5030  2          UAF$S_USERNAME, RECBUF[UAF$T_USERNAME]);
;  4978          5031  2 RECBUF[UAF$W_SALT] = 0;
;  4979          5032  2
;  4980          5033  2 !
;  4981          5034  2 !       Modify the RAB for our needs
;  4982          5035  2 !
;  4983          5036  2
;  4984          5037  2 UAFRAB[RAB$V_UIF] = 1;
;  4985          5038  2 UAFRAB[RAB$W_RSZ] = UAF$C_LENGTH;
;  4986          5039  2 UAFRAB[RAB$L_RBF] = RECBUF;
;  4987          5040  2
;  4988          5041  2 !
;  4989          5042  2 !       Modify the system password
;  4990          5043  2 !
```

```
;  4991          5044  2
;  4992          5045  2 $PUT(RAB=UAFRAB);
;  4993          5046  2
;  4994          5047  2 !
;  4995          5048  2 !       Replace the settings
;  4996          5049  2 !
;  4997          5050  2
;  4998          5051  2 UAFRAB[RAB$L_RBF] = .RBF;
;  4999          5052  2 UAFRAB[RAB$L_ROP] = .ROP;
;  5000          5053  2
;  5001          5054  1 end;


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

44  52  4F  57  53  53  41  50  5F  4D  45  54  53  59  53  0089C P.AFK:   .ASCII  \SYSTEM_PASSWORD\<0>                          ;
                                                        00  008AB                                                           ;
                                                    010E000F  008AC P.AFJ:   .LONG   17694735                               ;
                                                    00000000' 008B0          .ADDRESS P.AFK                                 ;
72  6F  77  73  73  61  50  2B  6D  65  74  73  79  53  3C  008B4 P.AFM:   .ASCII  \<System+Password>\<0><0><0>              ;
                                                00  00  00  3E  64  008C3                                                   ;
                                                    010E0011  008C8 P.AFL:   .LONG   17694737                               ;
                                                    00000000' 008CC          .ADDRESS P.AFM                                 ;
72  6F  77  73  73  61  50  2B  6D  65  74  73  79  53  3C  008D0 P.AFN:   .ASCII  \<System+Password>\<0><0><0>              ;
                                                00  00  00  3E  64  008DF                                                   ;


                                                                .PSECT  $CODE$,NOWRT,2

                                                03FC 00000         .ENTRY  UAF$MOD_SYS_PWD, Save R2,R3,R4,R5,R6,R7,R8,-; 4988
                                                                           R9                                               ;
                                            59 00000000'  00  9E 00002     MOVAB   P.AFJ, R9                                ;
                                            58 00000000'  00  9E 00009     MOVAB   UAFRAB+40, R8                            ;
                                            5E            04  C2 00010     SUBL2   #4, SP                                   ;
                                                02000000  8F  DD 00013     PUSHL   #33554432                                ; 5004
                                                      04  AE  D4 00019     CLRL    PWD+4                                    ;
                                            57            68  D0 0001C     MOVL    UAFRAB+40, RBF                           ; 5011
                                            56         DC A8  D0 0001F     MOVL    UAFRAB+4, ROP                            ; 5012
                                                     4200 8F  BB 00023     PUSHR   #^M<R9,SP>                               ; 5018
                                00000000G   00            02  FB 00027     CALLS   #2, CLI$GET_VALUE                        ;
                                                          6E  B5 0002E     TSTW    PWD                                      ; 5020
                                                          16  13 00030     BEQL    1$                                       ;
                                                       1C A9  9F 00032     PUSHAB  P.AFL                                    ; 5021
                                            7E            02  7D 00035     MOVQ    #2, -(SP)                                ;
                                                       0C AE  9F 00038     PUSHAB  PWD                                      ;
                                                     F940 C9  9F 0003B     PUSHAB  REC_ENCRYPT_DSC                          ;
                                00000000G   00            05  FB 0003F     CALLS   #5, LGI$HPWD                             ;
                                                          08  11 00046     BRB     2$                                       ;
08              00              6E                  00  2C 00048 1$:      MOVC5   #0, (SP), #0, #8, RECBUF+340             ; 5023
                                                   FB44 C8     0004D                                                        ;
20              20        24    A9                11  2C 00050 2$:      MOVC5   #17, P.AFN, #32, #32, RECBUF+4           ; 5030
                                                   F9F4 C8     00056                                                        ;
                                                   FB56 C8  B4 00059     CLRW    RECBUF+358                               ; 5031
                                            DC    A8        10  88 0005D     BISB2   #16, UAFRAB+4                            ; 5037
                                            FA    A8     0584 8F  B0 00061     MOVW    #1412, UAFRAB+34                         ; 5038
```

```
                                 68      F9F0    C8  9E 00067          MOVAB   RECBUF, UAFRAB+40                        ; 5039
                                           D8    A8  9F 0006C          PUSHAB  UAFRAB                                   ; 5045
                      00000000G  00            01  FB 0006F          CALLS   #1, SYS$PUT                              ;
                                 68            57  D0 00076          MOVL    RBF, UAFRAB+40                           ; 5051
                              DC 68            56  D0 00079          MOVL    ROP, UAFRAB+4                            ; 5052
                                                   04 0007D          RET                                              ; 5054

; Routine Size:  126 bytes,    Routine Base:  $CODE$ + 2A3A
```

```
;  5003          5055  1 %sbttl 'security_audit - perform a security audit'
;  5004          5056  1 routine security_audit (code, old_user) . novalue =
;  5005          5057  2 begin
;  5006          5058  2
;  5007          5059  2 !++
;  5008          5060  2 !
;  5009          5061  2 ! FUNCTIONAL DESCRIPTION:
;  5010          5062  2 !
;  5011          5063  2 !       Perform a security audit, if needed.
;  5012          5064  2 !
;  5013          5065  2 ! INPUTS:
;  5014          5066  2 !
;  5015          5067  2 !       CODE - Security audit record id
;  5016          5068  2 !       OLD_USER - Old username (optional depending on the record id)
;  5017          5069  2 !
;  5018          5070  2 ! IMPLICIT INPUTS:
;  5019          5071  2 !
;  5020          5072  2 !       PCB_STS - This process's PCB status
;  5021          5073  2 !       UAF$GQ_SYSUAFF - SYSUAF fields modified
;  5022          5074  2 !
;  5023          5075  2 ! OUTPUTS:
;  5024          5076  2 !
;  5025          5077  2 !       none
;  5026          5078  2 !
;  5027          5079  2 ! IMPLICIT OUTPUTS:
;  5028          5080  2 !
;  5029          5081  2 !       none
;  5030          5082  2 !
;  5031          5083  2 ! ROUTINE VALUE:
;  5032          5084  2 !
;  5033          5085  2 !       none
;  5034          5086  2 !
;  5035          5087  2 ! SIDE EFFECTS:
;  5036          5088  2 !
;  5037          5089  2 !       none
;  5038          5090  2 !
;  5039          5091  2 !--
;  5040          5092  2
;  5041          5093  2 external routine
;  5042          5094  2         nsa$event_audit;                ! Kernel mode auditing routine
;  5043          5095  2
;  5044          5096  2 external
;  5045          5097  2         nsa$gr_alarmvec : block [,byte], ! Security audit alarm vector
;  5046          5098  2         nsa$gr_journvec : block [,byte]; ! Security audit journal vector
;  5047          5099  2
;  5048          5100  2 macro
;  5049        M 5101  2         add_quad_packet(type, a_quad) = ! Add a quadword packet to list
;  5050        M 5102  2             begin
;  5051        M 5103  2             (.arglist_ptr)<0,16> = %name('nsa$k_pkttyp_', type);
;  5052        M 5104  2             arglist_ptr = .arglist_ptr + 2;
;  5053        M 5105  2             (.arglist_ptr)<0,16> = nsa$k_arg_mech_quad;
;  5054        M 5106  2             arglist_ptr = .arglist_ptr + 2;
;  5055        M 5107  2             (.arglist_ptr)<0,32> = .(a_quad)<0,32>;
;  5056        M 5108  2             arglist_ptr = .arglist_ptr + 4;
;  5057        M 5109  2             (.arglist_ptr)<0,32> = .(a_quad)<32,32>;
;  5058        M 5110  2             arglist_ptr = .arglist_ptr + 4;
;  5059        M 5111  2             arglist[nsa$b_arg_pktnum] = .arglist[nsa$b_arg_pktnum] + 1;
```

```
;  5060          5112  2              end%,
;  5061        M 5113  2          add_descr_packet(type, len, ptr) = ! Add a descriptor packet to list
;  5062        M 5114  2              begin
;  5063        M 5115  2              (.arglist_ptr)<0,16> = %name('nsa$k_pkttyp_', type);
;  5064        M 5116  2              arglist_ptr = .arglist_ptr + 2;
;  5065        M 5117  2              (.arglist_ptr)<0,16> = nsa$k_arg_mech_descr;
;  5066        M 5118  2              arglist_ptr = .arglist_ptr + 2;
;  5067        M 5119  2              (.arglist_ptr)<0,32> = len;
;  5068        M 5120  2              arglist_ptr = .arglist_ptr + 4;
;  5069        M 5121  2              (.arglist_ptr)<0,32> = ptr;
;  5070        M 5122  2              arglist_ptr = .arglist_ptr + 4;
;  5071        M 5123  2              arglist[nsa$b_arg_pktnum] = .arglist[nsa$b_arg_pktnum] + 1;
;  5072          5124  2              end%;
;  5073          5125  2
;  5074          5126  2 local
;  5075          5127  2          arglist         : block         ! Argument list for NSA$EVENT_AUDIT
;  5076          5128  2              [nsa$k_arghdr_length + ((2 + 2 + 8) * 6), byte],
;  5077          5129  2          arglist_ptr;                    ! Packet fill in pointer
;  5078          5130  2
;  5079          5131  2 ch$fill(0, %allocation(arglist), arglist); ! Clear out the argument list
;  5080          5132  2
;  5081          5133  2 if .nsa$gr_alarmvec[nsa$v_evt_uaf]       ! If alarm auditing,
;  5082          5134  2 then arglist[nsa$v_arg_flag_alarm] = true; !  perform alarm audit
;  5083          5135  2
;  5084          5136  2 if .nsa$gr_journvec[nsa$v_evt_uaf]       ! If journal auditing,
;  5085          5137  2 then arglist[nsa$v_arg_flag_journ] = true; !  perform journal audit
;  5086          5138  2
;  5087          5139  2 if .pcb_sts[$bitposition(pcb$v_secaudit)] ! If mandatory auditing,
;  5088          5140  2 then arglist[nsa$v_arg_flag_mandy] = true; !  perform mandatory audit
;  5089          5141  2
;  5090          5142  2 if .arglist[nsa$b_arg_flag] eql 0        ! If no audit requested,
;  5091          5143  2 then return;                             !  simply exit
;  5092          5144  2
;  5093          5145  2 arglist[nsa$l_arg_id] = .code;           ! Set audit record type/subtype
;  5094          5146  2
;  5095          5147  2 arglist_ptr = arglist[nsa$t_arg_list];  ! Address packet(s) in arg list
;  5096          5148  2
;  5097          5149  2 if .arglist[nsa$w_arg_type] eql nsa$k_rectyp_sysuaf
;  5098          5150  2 then                                     ! For SYSUAF records...
;  5099          5151  3     begin
;  5100          5152  3     if .arglist[nsa$w_arg_subtype] neq nsa$k_rectyp_sysuaf_del
;  5101          5153  3     then
;  5102          5154  3         add_quad_packet(sysuaff, uaf$gq_sysuaff);
;  5103        P 5155  3     add_descr_packet(usernam,
;  5104        P 5156  3                      namelen(uaf$s_username, recbuf[uaf$t_username]),
;  5105          5157  3                      recbuf[uaf$t_username]);
;  5106          5158  3     if .arglist[nsa$w_arg_subtype] eql nsa$k_rectyp_sysuaf_cop
;  5107          5159  3     or .arglist[nsa$w_arg_subtype] eql nsa$k_rectyp_sysuaf_ren
;  5108          5160  3     then
;  5109        P 5161  3         add_descr_packet(usernam,
;  5110        P 5162  3                          namelen(uaf$s_username, .old_user),
;  5111          5163  3                          .old_user);
;  5112          5164  3     end
;  5113          5165  2 else                                     ! For NETUAF records...
;  5114          5166  3     begin
;  5115        P 5167  3     add_descr_packet(nodenam,
;  5116        P 5168  3                      namelen(naf$s_node, netbuf[naf$t_node]),
```

```
;  5117          5169  3                          netbuf[naf$t_node]);
;  5118        P 5170  3      add_descr_packet(usernam,
;  5119        P 5171  3                          namelen(naf$s_remuser, netbuf[naf$t_remuser]),
;  5120          5172  3                          netbuf[naf$t_remuser]);
;  5121        P 5173  3      add_descr_packet(usernam,
;  5122        P 5174  3                          namelen(naf$s_localuser, netbuf[naf$t_localuser]),
;  5123          5175  3                          netbuf[naf$t_localuser]);
;  5124          5176  3      if .arglist[nsa$w_arg_subtype] eql nsa$k_rectyp_netuaf_mod
;  5125          5177  3      then
;  5126          5178  4          begin
;  5127        P 5179  4          add_descr_packet(nodenam,
;  5128        P 5180  4                           namelen(naf$s_node, netbuf[naf$t_node]),
;  5129          5181  4                           netbuf[naf$t_node]);
;  5130        P 5182  4          add_descr_packet(usernam,
;  5131        P 5183  4                           namelen(naf$s_remuser, netbuf[naf$t_remuser]),
;  5132          5184  4                           netbuf[naf$t_remuser]);
;  5133        P 5185  4          add_descr_packet(usernam,
;  5134        P 5186  4                           namelen(uaf$s_username, .old_user),
;  5135          5187  4                           .old_user);
;  5136          5188  3          end;
;  5137          5189  2      end;
;  5138          5190  2
;  5139          5191  2 arglist[nsa$l_arg_count] = (.arglist_ptr - (arglist + 4)) / 4; ! Set # args
;  5140          5192  2
;  5141          5193  2 $cmkrnl(routin = nsa$event_audit, arglst = arglist); ! Do the security audit
;  5143          5195  1 end;


                                                .EXTRN  NSA$EVENT_AUDIT
                                                .EXTRN  NSA$GR_ALARMVEC
                                                .EXTRN  NSA$GR_JOURNVEC
                                                .EXTRN  SYS$CMRNL

                                        007C 00000 SECURITY_AUDIT:
                                                .WORD   Save R2,R3,R4,R5,R6                     ; 5056
                        56 00000000'    00  9E 00002            MOVAB   NETBUF, R6
                        5E          AC  AE  9E 00009            MOVAB   -84(SP), SP
0054    8F      00          6E          00  2C 0000D            MOVC5   #0, (SP), #0, #84, ARGLIST      ; 5131
                                            6E  00014
                04 00000000G    00          02  E1 00015        BBC     #2, NSA$GR_ALARMVEC, 1$         ; 5133
                          08    AE          01  88 0001D        BISB2   #1, ARGLIST+8                   ; 5134
                04 00000000G    00          02  E1 00021 1$:    BBC     #2, NSA$GR_JOURNVEC, 2$         ; 5136
                          08    AE          02  88 00029        BISB2   #2, ARGLIST+8                   ; 5137
                04 00000000'    00          03  E1 0002D 2$:    BBC     #3, PCB_STS+3, 3$               ; 5139
                          08    AE          04  88 00035        BISB2   #4, ARGLIST+8                   ; 5140
                                      08    AE  95 00039 3$:    TSTB    ARGLIST+8                       ; 5142
                                            01  12 0003C        BNEQ    4$
                                            04  0003E          RET
                          04    AE      04  AC  D0 0003F 4$:    MOVL    CODE, ARGLIST+4                 ; 5145
                                52      0C  AE  9E 00044        MOVAB   ARGLIST+12, ARGLIST_PTR         ; 5147
                                02      04  AE  B1 00048        CMPW    ARGLIST+4, #2                   ; 5149
                                            3C  12 0004C        BNEQ    6$
                                02      06  AE  B1 0004E        CMPW    ARGLIST+6, #2                   ; 5152
                                            0F  13 00052        BEQL    5$
                                82 00030012 8F  D0 00054        MOVL    #196626, (ARGLIST_PTR)+         ; 5154
```

```
                                82      F9F8    C6  7D 0005B          MOVQ    UAF$GQ_SYSUAFF, (ARGLIST_PTR)+          ;
                                  09            AE  96 00060          INCB    ARGLIST+9                               ;
                                82 0004000A     8F  D0 00063 5$:      MOVL    #262154, (ARGLIST_PTR)+                 ; 5157
                          FA80  C6              20  3A 0006A          LOCC    #32, #32, RECBUF+4                      ;
                            82                  20  50 C3 00070      SUBL3   R0, #32, (ARGLIST_PTR)+                 ;
                                82      FA80    C6  9E 00074          MOVAB   RECBUF+4, (ARGLIST_PTR)+                ;
                                  09            AE  96 00079          INCB    ARGLIST+9                               ;
                                04        06    AE  B1 0007C          CMPW    ARGLIST+6, #4                           ; 5158
                                                  7D 13 00080          BEQL    8$                                      ;
                                05        06    AE  B1 00082          CMPW    ARGLIST+6, #5                           ; 5159
                                                  49 12 00086          BNEQ    7$                                      ;
                                                  75 11 00088          BRB     8$                                      ; 5163
                                82 00040009     8F  D0 0008A 6$:      MOVL    #262153, (ARGLIST_PTR)+                 ; 5169
                              66                  20  3A 00091          LOCC    #32, #32, NETBUF                        ;
                            82                  20  50 C3 00095      SUBL3   R0, #32, (ARGLIST_PTR)+                 ;
                                82              66  9E 00099          MOVAB   NETBUF, (ARGLIST_PTR)+                  ;
                                  09            AE  96 0009C          INCB    ARGLIST+9                               ;
                                82 0004000A     8F  D0 0009F          MOVL    #262154, (ARGLIST_PTR)+                 ; 5172
                          20    A6              20  3A 000A6          LOCC    #32, #32, NETBUF+32                     ;
                            82                  20  50 C3 000AB      SUBL3   R0, #32, (ARGLIST_PTR)+                 ;
                                82        20    A6  9E 000AF          MOVAB   NETBUF+32, (ARGLIST_PTR)+               ;
                                  09            AE  96 000B3          INCB    ARGLIST+9                               ;
                                82 0004000A     8F  D0 000B6          MOVL    #262154, (ARGLIST_PTR)+                 ; 5175
                          40    A6              20  3A 000BD          LOCC    #32, #32, NETBUF+64                     ;
                            82                  20  50 C3 000C2      SUBL3   R0, #32, (ARGLIST_PTR)+                 ;
                                82        40    A6  9E 000C6          MOVAB   NETBUF+64, (ARGLIST_PTR)+               ;
                                  09            AE  96 000CA          INCB    ARGLIST+9                               ;
                                03        06    AE  B1 000CD          CMPW    ARGLIST+6, #3                           ; 5176
                                                  43 12 000D1 7$:       BNEQ    9$                                      ;
                                82 00040009     8F  D0 000D3          MOVL    #262153, (ARGLIST_PTR)+                 ; 5181
                              66                  20  3A 000DA          LOCC    #32, #32, NETBUF                        ;
                            82                  20  50 C3 000DE      SUBL3   R0, #32, (ARGLIST_PTR)+                 ;
                                82              66  9E 000E2          MOVAB   NETBUF, (ARGLIST_PTR)+                  ;
                                  09            AE  96 000E5          INCB    ARGLIST+9                               ;
                                82 0004000A     8F  D0 000E8          MOVL    #262154, (ARGLIST_PTR)+                 ; 5184
                          20    A6              20  3A 000EF          LOCC    #32, #32, NETBUF+32                     ;
                            82                  20  50 C3 000F4      SUBL3   R0, #32, (ARGLIST_PTR)+                 ;
                                82        20    A6  9E 000F8          MOVAB   NETBUF+32, (ARGLIST_PTR)+               ;
                                  09            AE  96 000FC          INCB    ARGLIST+9                               ;
                                82 0004000A     8F  D0 000FF 8$:      MOVL    #262154, (ARGLIST_PTR)+                 ; 5187
                          08    BC              20  3A 00106          LOCC    #32, #32, @OLD_USER                     ;
                            82                  20  50 C3 0010B      SUBL3   R0, #32, (ARGLIST_PTR)+                 ;
                                82        08    AC  D0 0010F          MOVL    OLD_USER, (ARGLIST_PTR)+                ;
                                  09            AE  96 00113          INCB    ARGLIST+9                               ;
                                50        04    AE  9E 00116 9$:      MOVAB   ARGLIST+4, R0                           ; 5191
                                52              50  C2 0011A          SUBL2   R0, R2                                  ;
                              6E                52  04  C7 0011D      DIVL3   #4, R2, ARGLIST                         ;
                                                    5E  DD 00121      PUSHL   SP                                      ; 5193
                                    000000000G  00  9F 00123          PUSHAB  NSA$EVENT_AUDIT                         ;
                        00000000G   00              02  FB 00129      CALLS   #2, SYS$CMKRNL                          ;
                                                    04 00130          RET                                             ; 5195

; Routine Size:  305 bytes,    Routine Base:  $CODE$ + 2AB8
```

```
; 5145          5196  1 end                                   ! End of module
; 5146          5197  0 eludom


                                                                .EXTRN  LIB$STOP

;                                   PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $PLIT$                         2276  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $OWN$                          4364  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $GLOBAL$                       2032  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                        11241  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  .  ABS  .                         0  NOVEC,NOWRT,NORD ,NOEXE,NOSHR,  LCL,  ABS,  CON,NOPIC,ALIGN(0)


;                                 Library Statistics
;                                        -------- Symbols --------      Pages       Processing
;       File                             Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1          18619       300         1      1000          00:02.0


;                                   COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:UAFMAIN/OBJ=OBJ$:UAFMAIN MSRC$:UAFMAIN/UPDATE=(ENH$:UAFMAIN)

; Size:         11241 code + 8672 data bytes
; Run Time:       03:09.3
; Elapsed Time:   04:20.7
; Lines/CPU Min:   1646
; Lexemes/CPU-Min: 26564
; Memory Used:  654 pages
; Compilation Complete
```

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

UAFMAIN
LIS

RIGHTSMAN
LIS

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY